Robust classifcation methods on the space of covariance matrices. : application to texture and polarimetric synthetic aperture radar image classification

Ioana Ilea

▶ To cite this version:

Ioana Ilea. Robust classifcation methods on the space of covariance matrices. : application to texture and polarimetric synthetic aperture radar image classification. Other. Université de Bordeaux; Universitatea tehnică (Cluj-Napoca, Roumanie), 2017. English. NNT : 2017BORD0006 . tel-01511645

HAL Id: tel-01511645
https://theses.hal.science/tel-01511645

Submitted on 21 Apr 2017

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

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

THÈSE EN COTUTELLE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ET DE L'UNIVERSITÉ TECHNIQUE DE CLUJ-NAPOCA

École Doctorale Sciences Physiques et de l'Ingénieur
Spécialité : Automatique, Productique, Signal et Image, Ingénierie Cognitique

École Doctorale Électronique, Télécommunications et Technologies de l'Information
Spécialité : Ingénierie Électronique et Télécommunications

Par Ioana ILEA

Robust Classification Methods on the Space of Covariance Matrices. Application to Texture and Polarimetric Synthetic Aperture Radar Image Classification

Sous la direction de : M. Christian GERMAIN
Co-directeur : M. Romulus TEREBES
Co-encadrant : M. Lionel BOMBRUN

Soutenue le 26 janvier 2017

Membres du jury :

Mme. Monica BORDA	Professeur Université Technique de Cluj-Napoca	Président
M. Emmanuel TROUVE	Professeur Université Savoie Mont-Blanc	Rapporteur
M. Frédéric PASCAL	Professeur L2S / Centrale Supélec	Rapporteur
Mme. Isabelle CHAMPION	Chargée de recherche INRA	Examinateur
M. Mathieu FAUVEL	Maitre de conférences ENSAT	Examinateur
M. Christian GERMAIN	Professeur Bordeaux Sciences Agro	Directeur
M. Romulus TEREBES	Professeur Université Technique de Cluj-Napoca	Co-directeur
M. Lionel BOMBRUN	Maitre de conférences Bordeaux Sciences Agro	Co-encadrant

Titre: Classification robuste sur l'espace des matrices de covariance. Application à la texture et aux images de télédétection polarimétriques Radar à Ouverture Synthétique

Résumé: Au cours de ces dernières années, les matrices de covariance ont montré leur intérêt dans de nombreuses applications en traitement du signal et de l'image. Les travaux présentés dans cette thèse se concentrent sur l'utilisation de ces matrices comme descripteurs pour la classification. Dans ce contexte, des algorithmes robustes de classification sont proposés en développant les aspects suivants.

Tout d'abord, des estimateurs robustes de la matrice de covariance sont utilisés afin de réduire l'impact des observations aberrantes. Puis, les distributions Riemannienne Gaussienne et de Laplace, ainsi que leur extension au cas des modèles de mélange, sont considérés pour la modélisation des matrices de covariance. Les algorithmes de type k-moyennes et d'espérance-maximisation sont étendus au cas Riemannien pour l'estimation de paramètres de ces lois : poids, centroïdes et paramètres de dispersion. De plus, un nouvel estimateur du centroïde est proposé en s'appuyant sur la théorie des M-estimateurs : l'estimateur de Huber. En outre, des descripteurs appelés vecteurs Riemannien de Fisher sont introduits afin de modéliser les images non-stationnaires. Enfin, un test d'hypothèse basé sur la distance géodésique est introduit pour réguler la probabilité de fausse alarme du classifieur. Toutes ces contributions sont validées en classification d'images de texture, de signaux du cerveau, et d'images polarimétriques radar simulées et réelles.

Mots clés: Matrice de covariance, classification robuste, texture, espace Riemannien

Unité de recherche

Laboratoire IMS, CNRS UMR-5218, Groupe Signal et Image,
351 Cours de la Libération, 33400 Talence, France.

Université Technique de Cluj-Napoca,
71-73 Calea Dorobantilor, Cluj-Napoca, Roumanie.

Title: Robust Classification Methods on the Space of Covariance Matrices. Application to Texture and Polarimetric Synthetic Aperture Radar Image Classification

Abstract: In the recent years, covariance matrices have demonstrated their interest in a wide variety of applications in signal and image processing. The work presented in this thesis focuses on the use of covariance matrices as signatures for robust classification. In this context, a robust classification workflow is proposed, resulting in the following contributions.

First, robust covariance matrix estimators are used to reduce the impact of outlier observations, during the estimation process. Second, the Riemannian Gaussian and Laplace distributions as well as their mixture model are considered to represent the observed covariance matrices. The k-means and expectation maximization algorithms are then extended to the Riemannian case to estimate their parameters, that are the mixture's weight, the central covariance matrix and the dispersion. Next, a new centroid estimator, called the Huber's centroid, is introduced based on the theory of M-estimators. Further on, a new local descriptor named the Riemannian Fisher vector is introduced to model non-stationary images. Moreover, a statistical hypothesis test is introduced based on the geodesic distance to regulate the classification false alarm rate. In the end, the proposed methods are evaluated in the context of texture image classification, brain decoding, simulated and real PolSAR image classification.

Keywords: Covariance matrix, robust classification, texture, Riemannian space

---

Unité de recherche

Laboratoire IMS, CNRS UMR-5218, Groupe Signal et Image,
351 Cours de la Libération, 33400 Talence, France.

Université Technique de Cluj-Napoca,
71-73 Calea Dorobantilor, Cluj-Napoca, Roumanie.

Contents

Acknowledgments

First, I would like to express all my gratitude to my advisers, Prof. Christian Germain, Assoc. Prof. Lionel Bombrun and Prof. Romulus Terebes for making this thesis possible, for their support and guidance all along these three years. Thank you for your patience, encouragements, useful advises and remarks and for offering me three wonderful years.

Besides my advisers, I would like to thank my thesis committee Prof. Emmanuel Trouvé, Prof. Frédéric Pascal, Prof. Monica Borda, Assoc. Prof. Mathieu Fauvel and Mrs. Isabelle Champion for accepting to evaluate this work and for their valuable feedback.

My sincere thanks go to Prof. Monica Borda, for making possible my collaboration with the University of Bordeaux, and the Signal and Image Processing Group at the IMS Laboratory. Thank you also for offering me the possibility of being part of your research and teaching group at the Technical University of Cluj-Napoca.

I would like to thank Prof. Yannick Berthoumieu for receiving me in the Signal and Image Processing research group at the IMS Laboratory.

I also thank Mrs. Isabelle Champion for providing us with the PolSAR images used in this research.

I wish to warmly thank all my colleagues and friends at IMS for all the good moments spent together.

Special thanks go to my parents, my brother and Fabien, for their unconditional support and for always being there for me.

List of Acronyms

BIC	Bayesian Information Criterion
BKF	Bessel K Forms
BoRW	Bag of Riemannian Words
BoW	Bag of Words
CM	Center of Mass
db4	Daubechies 4 Filter
FP	Fixed Point Estimator
FV	Fisher Vectors
EM	Expectation Maximization Algorithm
GD	Geodesic Distance
GGD	Generalized Gaussian Distribution
GLCM	Gray-Level Co-occurrence Matrix
GMM	Gaussian Mixture Model
HH	Horizontal Transmitting, Horizontal Receiving Polarization Image
HV	Horizontal Transmitting, Vertical Receiving Polarization Image
IDAN	Intensity-Driven-Adaptive-Neighborhood Filter
KL	Kullback-Leibler Divergence
k-NN	k-Nearest Neighbor
LBP	Local Binary Pattern
LE-BoRW	Log-Euclidean Bag of Riemannian Words
LQQ	Linear Quadratic Quadratic Function
MAD	Median Absolute Deviation
MDM	Minimum Distance to Mean
MEG	Magnetoencephalography
MGD	Multivariate Gaussian Distribution
MGGD	Multivariate Generalized Gaussian Distribution
MLE	Maximum Likelihood Estimator
NSCM	Normalized Sample Covariance Matrix
PDE	Partial Differential Equation

PolSAR	Polarimetric Synthetic Aperture Radar
RCovD	Region Covariance Descriptors
RFV	Riemannian Fisher Vectors
RGD	Riemannian Gaussian Distribution
RLD	Riemannian Laplace Distribution
RMed	Riemannian Median
RMSE	Root-Mean Square Error
R-VLAD	Riemannian Vectors of Locally Aggregated Descriptors
SAR	Synthetic Aperture Radar
SCM	Sample Covariance Matrix
SIFT	Scale-Invariant Feature Transform
SRAD	Speckle Reduction Anisotropic Diffusion
SVM	Support Vector Machine
SIRV	Spherically Invariant Random Vectors
VH	Vertical Transmitting, Horizontal Receiving Polarization Image
VLAD	Vectors of Locally Aggregated Descriptors
VV	Vertical Transmitting, Vertical Receiving Polarization Image
WT	Wavelet Transform
WD	Wishart Distribution

Résumé Étendu

Contexte scientifique

Les travaux présentés dans cette thèse concernent les méthodes de classification robuste sur l'espace des matrices de covariance. Par conséquent, l'ensemble des travaux présentés dans ce manuscrit ont été menés autour de deux concepts centraux: « la géométrie des matrices de covariance » et « les algorithmes robustes » pour l'estimation et la classification.

Motivation et objectifs

Au cours de ces dernières années, les matrices de covariance ont montré leur intérêt dans des nombreuses applications en traitement du signal et de l'image, telles que la détection de cibles en imagerie radar [Greco *et al.* 2014, Chen *et al.* 2011, Yang *et al.* 2010, Barbaresco *et al.* 2013], la segmentation d'images médicales [de Luis-García *et al.* 2011], la détection de visages [Robinson 2005], la détection de véhicules [Mader & Reese 2012], ou encore la classification [Formont *et al.* 2011, Barachant *et al.* 2013, Said *et al.* 2015a, Faraki *et al.* 2015]. Dans le cas de la classification des signaux ou des images, les matrices de covariance sont utilisées afin de caractériser les dépendances spatiales, temporelles, spectrales, polarimétriques, etc. qui peuvent exister dans ce type de données. Les travaux présentés dans cette thèse se concentrent sur la modélisation de l'information texturale et polarimétrique, alors qu'une petite partie est consacrée à la classification de signaux MEG (magnéto-encéphalographie).

En outre, des algorithmes robustes de classification sont proposés afin de réduire l'influence des observations aberrantes sur la classification. L'apparition de ces observations peut être expliquée par la variabilité intrinsèque des données, par des erreurs de mesure, par des erreurs de modélisation, etc. Indépendamment de leurs origines, elles ont un impact négatif sur la classification, ce qui justifie la nécessité d'utiliser des algorithmes robustes.

En partant de ces observations, les objectifs de la thèse sont les suivants :

- Le développement d'outils de modélisation et de classification adaptés à la géométrie de l'espace des matrices de covariance.
- Le développement d'outils de modélisation et de classification robustes aux données aberrantes.
- La validation des méthodes proposées en classification des images ou des signaux.
- L'étude de l'influence du filtrage sur les performances de classification.

Contributions

Les matrices de covariance sont des descripteurs importants pour la classification des signaux et des images. Ainsi, dans cette thèse, des méthodes de modélisation adaptées à leur espace sont proposées et intégrées dans la classification. Le schéma de classification proposé est présenté dans la Figure 1. Il consiste en plusieurs étapes. Tout d'abord, des descripteurs, notamment des matrices de covariance, sont extraits et modélisés dans l'espace Riemannien. A partir de ce moment, les algorithmes de classification peuvent être mis en place, ou des descripteurs locaux peuvent être considérés pour caractériser l'image, tels que les approches sac de mots Riemanniens (BoRW) [Faraki *et al.* 2014], les vecteurs Riemanniens des descripteurs agrégés localement (R-VLAD) [Faraki *et al.* 2015], ou les vecteurs Riemanniens de Fisher (RFV).

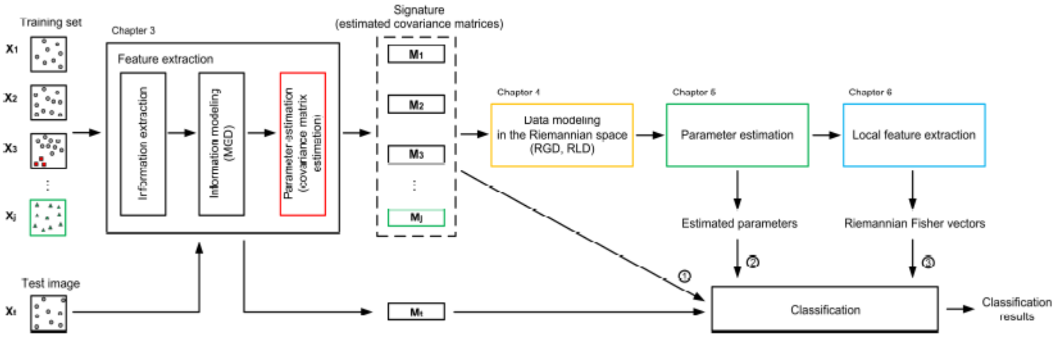


Figure 1: Le schéma de classification.

Dans le schéma de classification, l'aspect robuste intervient à plusieurs niveaux. Tout d'abord, une attention spéciale est donnée à l'estimation des matrices de covariance, qui doit être un processus robuste aux données aberrantes. En conséquence, des estimateurs robustes de ces matrices sont utilisés afin de réduire l'impact des observations aberrantes. Dans le contexte de la classification, la robustesse d'un algorithme concerne aussi la règle de décision. Afin de réguler la probabilité de fausse alarme du classifieur, il est possible de mettre en place un test d'hypothèse. En outre, la classification des matrices de covariance dans l'espace Riemannien implique la partition de l'ensemble de données dans des clusters. Cela nécessite l'estimation du centroïde de chaque cluster. A ce niveau, des algorithmes robustes aux matrices de covariance aberrantes sont impératifs. Enfin, pour la caractérisation des images non-stationnaires, les descripteurs globaux ne sont pas adaptés. Ainsi, des descripteurs locaux sont considérés, tels que les BoRW, les RFV, ou les R-VLAD.

Tous les aspects énumérés ci-dessus sont détaillés dans ce manuscrit, entrainant des contributions pour chaque étape du schéma de classification : extraction des descripteurs, modélisation des données, estimation des paramètres, extraction des descripteurs locaux et classification. Les contributions principales de la thèse peuvent être résumées de la façon suivante :

1. Extraction des descripteurs :
    - L'utilisation d'estimateurs robustes de la matrice de covariance. L'objectif de ces estimateurs est de réduire l'impact des données aberrantes sur la modélisation de l'information.
    - L'introduction d'une méthode de filtrage basée sur la diffusion anisotrope pour la réduction du bruit de speckle. L'objectif de cette étape est d'étudier le potentiel du filtrage sur la classification des images Polarimétrique Radar à Synthèse d'Ouverture (PolSAR).
2. Modélisation des données :
    - L'introduction de la distribution Riemannienne de Laplace, ainsi que son extension au cas des modèles de mélange pour la modélisation des matrices de covariance.
    - L'extension des algorithmes de type k-moyennes et d'espérance-maximisation (EM) au cas Riemannien pour l'estimation des paramètres de la loi Riemannienne de Laplace.
3. Estimation des paramètres :
    - L'introduction du centroïde de Huber. En s'appuyant sur la théorie des M-estimateurs, le centroïde de Huber est défini comme l'estimateur de la valeur centrale d'un ensemble de matrices de covariance.
    - Le calcul automatique du paramètre régulant le compromis entre robustesse et efficacité du centroïde de Huber. Afin de fixer ce paramètre, un algorithme basé sur le concept de MAD (*median absolute deviation*) est introduit.
4. Extraction de descripteurs locaux :
    - La définition des descripteurs appelés vecteurs Riemanniens de Fisher pour les distributions Riemannienne Gaussienne et de Laplace. Afin de modéliser les images non-stationnaires, les vecteurs de Fisher sont étendus au cas Riemannien en utilisant les densités de probabilité des lois Riemanniennes Gaussienne et de Laplace.
    - L'illustration du lien entre les vecteurs Riemanniens de Fisher et les vecteurs Riemanniens des descripteurs agrégés localement (R-VLAD).
5. Classification :
    - La définition d'un test d'hypothèse basé sur la distance géodésique pour la régulation de la fausse alarme du classifieur.
    - La validation de l'ensemble des méthodes proposées, dans le cadre de la classification de textures, des signaux MEG, ainsi que des images PolSAR simulées et réelles.

Chacun de ces sujets est abordé dans un chapitre distinct de la thèse, ce qui donne la structure suivante pour le manuscrit.

Contenu de la thèse

Chapitre 2 : Les textures dans le traitement d'images

Ce chapitre présente une introduction sur les textures et leur modélisation en traitement d'images.

A partir de la complexité de ce concept, des définitions de l'état de l'art sont présentées, ainsi que certaines méthodes pour l'extraction de l'information texturale. Une classification de ces méthodes est réalisée en les regroupant en deux classes : des méthodes basées sur l'analyse statistique de l'organisation spatiale des niveaux de gris et des méthodes stochastiques. La première catégorie inclue les matrices de co-occurrence des niveaux de gris [Haralick *et al.* 1973], les fonctions d'autocorrélation [Tuceryan & Jain 1993], les variogrammes [Matheron 1963, Curran 1988], les motifs binaires locaux (*Local Binary Patterns* - LBP) [Ojala *et al.* 1996], alors que la deuxième catégorie regroupe des méthodes basées sur le filtrage fréquentiel de l'image (le filtre de Gabor [Turner 1986,Jain & Farrokhnia 1991], la décomposition en ondelettes [Mallat 1989]) et la modélisation de ces coefficients extraits par des modèles probabilistes (la distribution Gaussienne généralisée [Do & Vetterli 2002], la distribution Gamma [Mathiassen *et al.* 2002], distribution Gaussienne généralisée multivariée [Verdoolaege & Scheunders 2011], les processus sphériquement invariant SIRV [Yao 1973,Gini & Greco 2002,Pascal *et al.* 2006,Vasile *et al.* 2010], la théorie des copules [Kwitt *et al.* 2009,Lasmar & Berthoumieu 2014]).

Les travaux présentés dans cette thèse se concentrent sur l'utilisation de la distribution Gaussienne multivariée de moyenne nulle pour caractériser les coefficients de la décomposition en ondelettes. Ce choix a été basé sur les propriétés intéressantes de la distribution, notamment la forme explicite de la distance géodésique. Cette distribution a un paramètre unique, qui est la matrice de covariance. Pour décrire l'espace de ces matrices, des modèles stochastiques ont été proposés dans la littérature, tels que la distribution de Wishart [Wishart 1928], ou les modèles de mélange d'échelle de Wishart [Lee *et al.* 1993, Freitas *et al.* 2003, Bombrun & Beaulieu 2008, Bombrun *et al.* 2011a]. Bien que ces modèles puissent être utilisés de manière efficace, ils ne prennent pas en compte la géométrie intrinsèque des données. Pour cela, les distributions Riemanniennes [Said *et al.* 2015b, Hajri *et al.* 2016] sont considérées dans cette thèse.

Chapitre 3 : Classification robuste sur l'espace des matrices de covariance

Ce chapitre présente l'extraction des descripteurs afin de modéliser l'information contenue dans les données et le développement d'un schéma de classification robuste. La première étape dans le schéma proposé est représentée par l'extraction de l'information texturale en utilisant une décomposition multi-échelles de l'image. Pour cela, la décomposition en ondelettes (WT) est considérée [Do & Vetterli 2002]. En classification, le principe général réside dans la modélisation des coefficients de détails par des densités de probabilités univariées [Do & Vetterli 2002], ou multivariées [Bombrun *et al.* 2011b,Verdoolaege & Scheunders 2012,Kwitt & Uhl 2010,Stitou *et al.* 2009]. Dans cette thèse, la distribution Gaussienne multivariée de moyenne nulle (MGD) est choisie et son paramètre, qui est la matrice de covariance, donne la signature finale de l'image. Afin de mettre en place des algorithmes robustes, le choix de l'estimateur de la matrice de covariance est très important. Ainsi, la classe des M-estimateurs [Huber 1964, Tyler 1987] ainsi que l'estimateur du point fixe [Tyler 1987], ont été introduit dans le contexte de l'estimation robuste, pour fonctionner correctement en présence d'observations aberrantes dans le jeux de données (marquées en rouge dans la Figure 1). En outre, un nouveau test d'hypothèse basé sur la distance géodésique est mis en place afin d'obtenir la régulation de la probabilité de fausse alarme du classifieur. La robustesse de ce classificateur est validée sur des données synthétiques ainsi que sur des données PolSAR simulées et réelles pour la classification de parcelles forestières.

Les expériences réalisées sur les données PolSAR ont eu plusieurs objectifs : l'évaluation de la statistique du test proposé, l'analyse de l'influence des paramètres d'acquisition (angle d'incidence, résolution spatiale, nombre de canaux polarimétriques) sur la classification, et l'introduction de méthodes capable d'exploiter les dépendances qui existent dans ces images (spatiale (S), polarimétrique (Polar)).

Le tableau 1 montre les performances de classification en termes de taux de bonne classification pour les approches proposées.

Méthode de classification	Taux de bonne classification
GLCM HV	$\mathbf{86.6 \pm 5.6}$
MGD HH + WT + S	59.0 ± 5.4
MGD Polar	$\mathbf{84.0 \pm 4.4}$
MGD Polar + WT	81.8 ± 4.0
MGD Polar + WT + S	63.5 ± 4.9

Tableau 1: Comparaison entre les algorithmes de classification pour les images SAR réelles en bande L. Les matrices de covariance sont estimées ici par maximum de vraisemblance.

Par ailleurs, les images de télédétection ont un niveau de bruit élevé ce qui représente le principal frein à leur utilisation. Ce bruit est pris en compte dans la modélisation stochastique multi-échelles utilisée, mais une réduction préalable du bruit, tout en préservant les caractéristiques géométriques des éléments texturaux, constitue une alternative de nature à améliorer les performances de classification des images polarimétriques SAR. Une méthode de filtrage basée sur les algorithmes de diffusion anisotrope et sur les équations aux dérivées partielles (EDP) est également proposée et validée sur des images PolSAR.

Le tableau 2 présente les performances de classification en termes de taux de bonne classification pour l'utilisation d'un seul canal polarimétrique. Les résultats sont comparés avec des autres algorithmes de filtrage (Gaussien, Boxcar et SRAD) et illustrent l'amélioration apportée par la méthode proposée.

Méthode de classification	Image originale	Image filtrée			
		Gaussien	Boxcar	SRAD	EDP
MGD HH + WT + S	57.94 ± 6.15	63.00 ± 4.09	62.28 ± 4.24	63.03 ± 5.14	65.47 ± 2.99
MGD HV + WT + S	61.09 ± 5.32	61.38 ± 3.94	62.88 ± 4.64	60.25 ± 6.05	64.47 ± 3.37
MGD VV + WT + S	59.66 ± 4.68	60.94 ± 5.66	65.50 ± 4.68	61.58 ± 5.20	65.91 ± 4.26

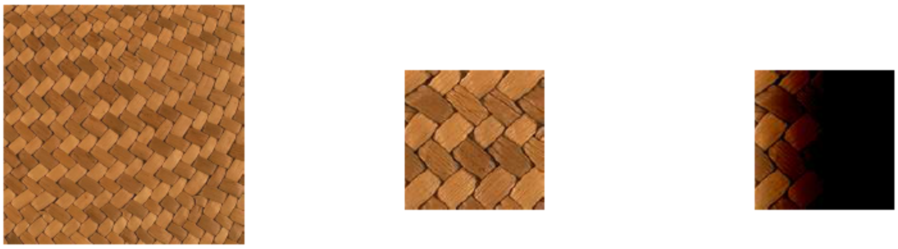
Tableau 2: Comparaison entre les performances de classification obtenues sur les images SAR en bande L non filtrées et filtrées.

Chapitre 4 : Distributions Riemanniennes dans l'espace des matrices de covariance

Ce chapitre est focalisé sur les distributions Riemanniennes pour la caractérisation de l'espace des matrices de covariance. Le chapitre commence avec une partie théorique sur la géométrie Riemannienne et introduit les lois Riemanniennes Gaussienne (RGD) [Said *et al.* 2015b] et de Laplace (RLD) [Hajri *et al.* 2016]. Ces densités de probabilité sont caractérisées par deux paramètres : le centroïde $\bar{\mathbf{M}}$ et le paramètre de dispersion σ. Sachant que pour la distribution Gaussienne Riemannienne le centroïde est donné par le centre de masse, ce modèle peut être influencé par les valeurs aberrantes. Pour résoudre ce problème, la distribution Riemannienne de Laplace est introduite, ayant la médiane Riemannienne comme valeur centrale. Afin de caractériser la diversité intra-classe naturellement présente dans les données, ces modèles sont généralisés au cas des modèles de mélange. L'estimation des paramètres du modèle de mélange est réalisée en utilisant des algorithmes classiques de type k-moyennes et d'espérance-maximisation étendus au cas Riemannien. Les deux modèles stochastiques sont comparés pour la classification de textures.

L'objectif de la partie expérimentale est d'analyser le comportement de ces modèles (RGD, RLD, ainsi que la distribution de Wishart (WD) [Lee *et al.* 1999, Saint-Jean & Nielsen 2013]) sur des données qui contiennent des valeurs aberrantes. Pour ces tests, une version modifiée de la base de texture VisTex [Vis] est considérée. Cette nouvelle base a été créée en ajoutant des patchs aberrants pour chacune de ses 40 classes de textures. Un exemple de texture, un de ses patchs et un patch aberrant sont montrés dans la Figure 2.

Les performances de classification sont présentées dans la Figure 3, sachant que l'estimation de paramètres a été réalisée en utilisant l'algorithme EM. Le nombre de clusters par classe a été fixé, ou calculé avec le critère d'information bayésien (BIC). Les résultats obtenus ont montré que le mélange de RLDs combiné avec le critère BIC pour l'estimation du nombre de clusters permet d'améliorer les performances de classification.



(a) (b) (c)

Figure 2: (a) Exemple de texture de la base VisTex, (b) un de ses patchs et (c) un patch aberrant.

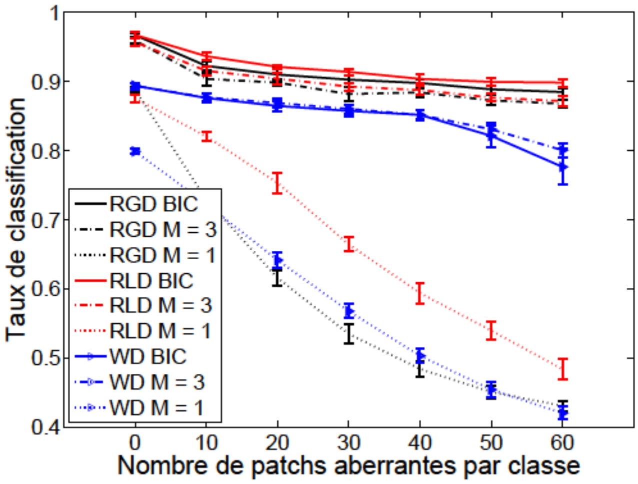


Figure 3: Performances de classification.

Chapitre 5 : Méthode robuste d'estimation du centroïde dans l'espace des matrices de covariance

Ce chapitre porte sur l'étude de la robustesse des estimateurs du centroïde d'un ensemble de matrices de covariance. A ce niveau, des estimateurs robustes du centroïde sont essentiels afin de prendre en compte les valeurs aberrantes découlant de la variabilité inhérente des données ou des mesures erronées. Ces observations aberrantes (représentées en vert dans la Figure 1) conduisent à des matrices de covariance estimées "atypiques" (représentées aussi en vert dans la Figure 1) qui doivent être prise en compte par les estimateurs robustes du centroïde. Dans la littérature, différents estimateurs ont été définis dans l'espace Riemannien, tels que le centre de masse, la médiane, ou les approches de type « *trimming* ». Dans ce chapitre, une nouvelle méthode pour l'estimation du centroïde est proposée en s'appuyant sur la théorie des M-estimateurs, et plus particulièrement sur la fonction de coût de Huber. L'estimateur de Huber proposé représente un compromis entre l'efficacité du centre de masse et la robustesse de la médiane. Ce compromis est contrôlé par le paramètre scalaire de la fonction de coût de Huber. Afin de fixer ce paramètre, un algorithme basé sur le concept de MAD (*median absolute deviation*) est proposé. Différentes expérimentations sur des données réelles et simulées sont présentées afin d'évaluer les performances d'estimation de l'estimateur de Huber pour le centroïde $\bar{\mathbf{M}}$. Deux applications en classification d'images texturées et en classification des signaux MEG sont proposées, illustrant l'intérêt de l'estimateur proposé.

Par exemple, les résultats obtenus pour la classification de signaux MEG sont montrés dans le tableau 3. Pour cette expérience, la base proposée pour la compétition « Biomag 2014 Decoding Challenge: Brain Decoding Across Subjects (DecMeg 2014)» [Dec] a été utilisée. L'objectif de ce test a été la prédiction d'un stimulus montré à un sujet, en utilisant l'activité cérébral. La base de données a été construite en ayant comme stimulus des images représentant des visages et des faux visages. En termes de traitement du signal, ceci revient à résoudre un problème de classification en deux classes. Dans ce contexte, la distribution Riemannienne Gaussienne a été considérée pour la modélisation de données, et les résultats ont été comparés avec la méthode MDM (*minimum distance to mean*) qui a remporté la compétition en 2014 [Barachant 2014, Barachant *et al.* 2012]. Les centroïdes ont été estimés en utilisant le centre de masse (CM) [Karcher 1977, Nielsen & Bhatia 2012, Fiori 2009], la médiane (Med) [Fletcher *et al.* 2009, Yang *et al.* 2010], l'estimateur de Huber (Huber), et les méthodes de type « *trimming* » (Trim) [Uehara *et al.* 2016]. Avec l'estimateur de Huber un gain a été obtenu lorsque le seuil est fixé à $T = 0.2$. La valeur calculée automatiquement pour le seuil donne une idée sur son ordre de grandeur. En ajustant cette valeur, de meilleures performances peuvent être obtenues.

Tableau 3: Classification des signaux du cerveau.

Estimateur	RGD	MDM
CM	73.845	**74.106**
Med	74.150	73.627
Huber $T = 0.2$	**75.109**	**74.847**
Huber $T = 0.5$	73.976	74.063
Huber $T = auto$	74.106	74.455
CM($\text{Trim}_\alpha^{\text{mean}}$)	73.888	73.976
CM($\text{Trim}_\alpha^{\text{med}}$)	74.237	73.801
Med($\text{Trim}_\alpha^{\text{mean}}$)	74.542	74.412
Med($\text{Trim}_\alpha^{\text{med}}$)	74.586	74.237

En outre, le potentiel des estimateurs robustes de la matrice de covariance et du centroïde d'un ensemble de matrices de covariance est analysé pour la classification.

Chapitre 6 : Vecteurs Riemanniens de Fisher

Ce chapitre présente une alternative à la classification basée sur des descripteurs globaux détaillée dans les chapitres antérieurs. Parfois, ces descripteurs ne sont pas adaptés pour prendre en compte les informations contenues dans les signaux ou les images. C'est par exemple le cas pour des signaux non-stationnaires. Pour résoudre ce problème, des méthodes de classification basées sur des descripteurs locaux sont proposées dans ce chapitre. Ainsi, des approches de type sac de mots, vecteurs de Fisher, ou vecteurs de descripteurs agrégés localement (VLAD) sont considérés. Récemment, les approches sac de mots et VLAD ont été généralisées au cas de descripteurs vivant sur une variété Riemannienne [Faraki *et al.* 2015]. Jusqu'à présent, les vecteurs de Fisher n'ont pas été généralisés de la même manière dû au manque d'un modèle probabiliste adapté aux descripteurs paramétriques. Dans ce chapitre, grâce au formalisme des lois Gaussiennes [Said *et al.* 2015b] et de Laplace [Hajri *et al.* 2016] sur des variétés Riemanniennes, la définition de ces descripteurs est proposée et les résultats obtenus sont validés sur des bases d'images texturées.

Par exemple, le tableau 4 montre les performances de classification obtenues sur la base VisTex en utilisant les descripteurs BoRW, RFV et R-VLAD. Les résultats montrent que pour cette expérience, l'utilisation de la loi de Laplace Riemannienne (RLD) apporte une petite amélioration en termes de taux de bonne classification. Les gains les plus importants, d'environ 7% et 4%, sont marquées en bleu. En outre, l'approche RFV proposée conduit à de meilleures performances que les methodes de l'état de l'art: BoRW [Faraki *et al.* 2014] et R-VLAD [Faraki *et al.* 2015].

Tableau 4: Résultats de classification obtenus sur la base VisTex en termes de taux de bonne classification.

Méthode	**Homosced.**	**Poids**	**RGD**	**RLD**
BoRW	non	oui	87.22 ± 1.19	87.70 ± 1.75
BoRW	non	non	87.51 ± 0.92	88.10 ± 1.42
BoRW [Faraki *et al.* 2014]	oui	non	87.20 ± 1.55	87.69 ± 0.93
BoRW	oui	oui	76.67 ± 2.35	69.01 ± 5.39
RFV : ϖ	non	oui	89.21 ± 0.94	90.11 ± 0.58
RFV : σ	non	oui	81.42 ± 1.12	88.51 ± 0.87
RFV : $\bar{\mathbf{M}}$	non	oui	87.22 ± 1.15	87.71 ± 1.06
RFV : σ, ϖ	non	oui	81.80 ± 0.60	85.36 ± 0.86
RFV : $\bar{\mathbf{M}}, \varpi$	non	oui	88.13 ± 0.67	88.45 ± 0.79
RFV : $\bar{\mathbf{M}}, \sigma$	non	oui	90.41 ± 0.86	91.07 ± 0.53
RFV : $\bar{\mathbf{M}}, \sigma, \varpi$	non	oui	89.93 ± 0.53	89.77 ± 1.13
R-VLAD [Faraki *et al.* 2015]	oui	non	87.94 ± 0.58	87.38 ± 0.73

Chapitre 7 : Conclusions et perspectives

Ce chapitre synthétise les principales conclusions de cette thèse et présente les perspectives pour les travaux futurs.

L'objectif principal de cette thèse a concerné le développement d'algorithmes de classification robustes, basés sur l'utilisation de matrices de covariance comme descripteurs de l'information texturale.

Dans ces travaux, les observations sont modélisées par une distribution Gaussienne multivariée de moyenne nulle. Cette loi de probabilité a un paramètre unique qui est la matrice de covariance et qui représente le descripteur utilisé en classification. L'étape de classification peut être implémentée en utilisant directement ces matrices, ou en les modélisant dans l'espace où vivent ces matrices de covariance qui est une variété Riemannienne. Dans le premier cas, un test d'hypothèse basé sur la distance géodésique et l'estimateur du point fixe a été proposé afin d'obtenir une règle de décision, qui permet de réguler la probabilité de fausse alarme. Dans l'espace des matrices de covariance, les modèles de mélange de loi Riemanniennes Gaussienne ou de Laplace peuvent être considérées. Dans ce cas, l'ensemble de données est caractérisé par une valeur centrale et un paramètre de dispersion pour chaque mode du modèle de mélange. Ces paramètres peuvent être estimés par des algorithmes de type k-moyennes et d'espérance-maximisation. Le calcul de la valeur centrale doit être robuste aux données aberrantes. Ainsi, un algorithme basé sur la théorie des M-estimateurs a été proposé.

En outre, des descripteurs locaux de type vecteurs de Fisher ont été généralisés au cas des matrices de covariance qui vivent dans une variété Riemannienne.

Les algorithmes ont été validés pour la classification de textures, de signaux MEG, d'images PolSAR simulées et réelles. Pour la dernière application, des algorithmes de filtrage ont été introduit, afin de réduire le bruit de speckle inhérent aux images radar, tout en préservant les structures présentes dans ces images.

Les travaux présentés dans cette thèse ouvrent la voie à plusieurs perspectives de travaux :

- La généralisation des méthodes proposées aux modèles statistiques qui ne sont pas Gaussiens. En effet, nous nous sommes intéressés ici uniquement à l'espace des matrices de covariance. Comme exposé dans le chapitre 2, d'autres modèles peuvent être considérés pour décrire les observations comme les modèles SIRV, les MGGD, les copules, ... Une perspective serait donc d'étendre nos travaux à ces modèles;

- Dans cette thèse, la plupart des outils proposés (définition de lois sur des variétés) n'est valable que pour des matrices de covariance réelles. Une piste à explorer serait d'étendre ces travaux au cas des matrices de covariance complexes;

- Le développement de lois sur des variétés Riemanniennes adaptés à l'espace des matrices de covariance structurées (par exemple, matrices Toeplitz, bloc Toeplitz, ...);

- Dans les approches BoRW, R-VLAD et RFV, la répartition spatiale des patchs n'a pas été prise en compte. Il pourrait être intéressant d'exploiter cette information pour améliorer les performances de classification. Pour cela, nous pourrons proposer une approche similaire aux matrices de co-occurrences des niveaux de gris utilisés classiquement en analyse de texture: les matrices de co-occurrences des covariances.

Contexte administratif

Les travaux présentés dans cette thèse ont été menés dans le cadre d'un contrat de cotutelle entre l'Université de Bordeaux en France et l'Université Technique de Cluj-Napoca en Roumanie. La complémentarité des deux équipes de recherche a permis l'exploitation de différents domaines de recherche comme la modélisation texturale et la restauration d'image fondée sur les EDP. En outre, la thèse a fait partie d'un projet de recherche international financé par le ministère des affaires étrangères et du développement international, ainsi que par l'agence exécutive pour l'enseignement supérieur, la recherche, le développement et l'innovation dans le cadre du projet numéro 32619VL (France) et PNII Capacitati 779/27.06.2014 (Roumanie). De plus, la thèse a été financée par le Centre Nationale de la Recherche Scientifique et Bordeaux Sciences Agro.

Chapter 1

Introduction

Contents

1.1 Scientific Context

The main topic of this thesis is the use of covariance matrices as signal and image signatures for robust classification. Therefore, the entire work presented here is built around two central concepts, that are "covariance matrices" and "robust classification algorithms". The interest on these subjects is explained next.

1.1.1 Motivation and Objectives

In the recent years, covariance matrices have demonstrated their importance in a wide variety of applications in signal and image processing, being related to array processing [Ollila & Koivunen 2003], radar detection [Greco *et al.* 2014, Chen *et al.* 2011, Yang *et al.* 2010, Barbaresco *et al.* 2013], medical image segmentation [de Luis-García *et al.* 2011], face detection [Robinson 2005], vehicle detection [Mader & Reese 2012], or classification [Formont *et al.* 2011,Barachant *et al.* 2013, Said *et al.* 2015a,Faraki *et al.* 2015a]. In the context of signal and image classification, covariance matrices can be used to model different kinds of dependence, like spatial, temporal, spectral, polarimetric dependence, etc. The work presented in this thesis focuses almost entirely on texture and polarimetric information modeling. A small part is dedicated to magnetoencephalography (MEG) data.

In addition, robust algorithms are desired in order to reduce the influence of outliers on the classification results. The presence of outliers may be explained by the inherent variability of data, by faulty measurements, by errors in the modeling process, etc. Independent of their source, they have a negative impact on the final results, motivating the need of robust algorithms.

Considering these aspects, this thesis has several objectives:

- To develop modeling and classification tools adapted to the particular geometry of the space of covariance matrices.
- To develop modeling and classification tools robust to outliers.
- To evaluate the performance of the proposed methods on signal and image classification.
- To study the impact of image filtering on classification results.

1.1.2 Contributions

Since covariance matrices are important features for signal and image classification, appropriate methods able to deal with the properties of their space are introduced and integrated into a classification workflow. The proposed workflow is presented in Figure 1.1 and it contains the following steps. First, features or more precisely covariance matrices are extracted and then, they are modeled on the Riemannian manifold. Starting from this point, classification algorithms can be directly applied, or local feature based methods can be considered to encode the image, like bag of Riemannian words (BoRW) [Faraki *et al.* 2014], Riemannian Fisher vectors (RFV), or Riemannian vectors of locally aggregated descriptors (R-VLAD) [Faraki *et al.* 2015a].

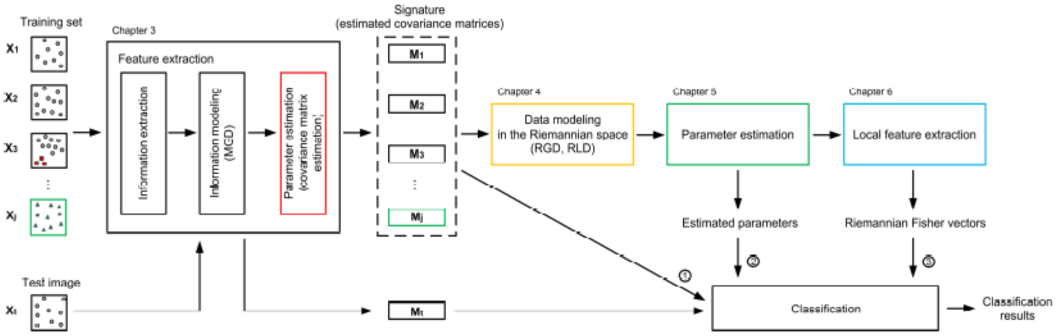


Figure 1.1: Classification workflow.

In this workflow, the idea of robustness appears at several levels. First, a special care is required for the covariance matrix estimation process. More precisely, the covariance matrix estimation has to be able to deal with the outlier values present in the observations' structure. Second, in the classification context, the robustness of an algorithm concerns also the decision-making strategies during the classification step. For that, a statistical hypothesis test can be considered to regulate the false alarm rate. In addition, when modeled in the Riemannian space, the covariance matrix classification implies the data's partition into clusters, and therefore the computation of some central values. At this stage, the centroid estimation has to be robust to the possible aberrant covariance matrices. In the end, for non-stationary images (such as local deformation), global descriptors are not adapted. Therefore, local based descriptors should be considered, such as BoRW, R-VLAD and RFV.

All the previously mentioned aspects are addressed in this thesis, resulting in the contributions listed and described below for each stage of the proposed workflow: feature extraction, data modeling, parameter estimation, local feature extraction and classification. The main contributions of this PhD thesis can be summarized as follows:

Feature extraction:

- Use of robust covariance matrix estimators for information modeling. The purpose of these estimators is to reduce the impact of the outlier observations, during the information modeling process.

- Introduction of a directional diffusion based denoising method, for speckle reduction. The objective of this step is to study the potential of filtering, prior to the classification of Polarimetric Synthetic Aperture Radar (PolSAR) images.

Data modeling:

- Introduction of the Riemannian Laplace distribution along with the mixture model for covariance matrix modeling.
- Extension of the k-means and expectation maximization algorithms for parameter estimation.

Parameter estimation:

- Introduction of the Huber's centroid. Based on the theory of M-estimators, the Huber's centroid is defined to estimate the central element from a set of N covariance matrices.
- Automatic computation of the Huber's centroid parameter. The Huber's centroid has a unique parameter, that is the threshold value discriminating between normal and aberrant data. Based on the concept of median absolute deviation (MAD) extended to the case of covariance matrices, a method to automatically tune the threshold's value is introduced.

Local feature extraction:

- Definition of the Riemannian Fisher vectors for the Riemannian Gaussian and Laplace mixture models. To address the problem of non-stationary image classification, the Fisher vectors are extended to the Riemannian case, based on the Riemannian Gaussian and Laplace probability density functions.
- Illustration of the relation between the Riemannian Fisher vectors and the Riemannian vectors of locally aggregated descriptors.

Classification:

- Definition of a statistical hypothesis test based on the geodesic distance to regulate the false alarm rate.
- Evaluation of the proposed methods in the context of texture image classification, brain decoding, simulated and real PolSAR image classification.

Further on, each of these topics is addressed in a distinct chapter, giving the following structure for the present work.

1.2 Thesis Outline

The remainder of this thesis is structured as follows.

Chapter 2 represents an introduction on textures and their modeling in image processing. Starting from the complexity of this concept, state-of-the-art definitions are presented, along with some methods for transforming the textural information into descriptors, or features, used in computer vision. A classification of the feature extraction methods is made, by grouping them into two categories: methods based on descriptive statistics and methods based on stochastic modeling. Further on, a special attention is given to covariance matrices and their ability to model textures and more generally, images, or videos.

Chapter 3 deals with textural feature extraction and its integration into a robust classification workflow. The first step in the proposed workflow consists in extracting the textural information by means of a multiscale decomposition. For this purpose, the wavelet decomposition is used. Once extracted, the wavelet coefficients are modeled by multivariate distributions to capture the dependencies existing in the image. More precisely, the zero-mean multivariate Gaussian distribution is used and its parameter, that is the covariance matrix, gives the final texture's signature. In order to obtain robust classification algorithms, the choice of the covariance matrix estimator is very important, knowing that the estimation process has to be able to deal with the outlier values (marked by red squares in Figure 1.1) present in the observations' structure. Therefore, the fixed point estimator and the class of M-estimators are studied. Further on, a hypothesis test based on the geodesic distance is introduced to regulate the false alarm and its noise robustness and classification efficiency are studied. The obtained statistic is applied next to PolSAR image classification. Several experiments are designed, to study the influence of the acquisition parameters. In the end, the classification workflow is modified by introducing a directional diffusion based filtering preprocessing stage to reduce the speckle noise present in PolSAR data. This algorithm, based on the partial differential equation formalism, is defined and applied for synthetic and real PolSAR data.

Chapter 4 focuses on the Riemannian distributions for modeling the space of covariance matrices. This chapter begins with a short theoretical part on the Riemannian geometry and then, it introduces the Riemannian Gaussian (RGD) and Laplace (RLD) distributions. These distributions are characterized by two parameters: the central value, and the dispersion around it. Knowing that for the RGD the central value is the Riemannian center of mass, the model may be influenced by the outliers present in the data. To overcome this problem, the Riemannian Laplace distribution, for which the centroid is the robust Riemannian median, is introduced. For each distribution, several elements are detailed: the probability density function, the mixture model and the parameter estimation methods. In the end, the distributions are compared in the context of texture image classification, where their purpose is to model the within-class diversity.

Chapter 5 follows the idea of studying the robustness of centroid estimation methods for covariance matrices. At this point, robust centroid estimators are essential to deal with outliers arising from the inherent variability of the data, or from faulty measurements. These aberrant observations (marked in green in Figure 1.1) give aberrant estimated covariance matrices (also marked in green in Figure 1.1) that have to be identified by the robust estimators of central values. In this context, the center of mass, the median and the geometric trimmed averages defined on the Riemannian manifold are analyzed, by presenting their advantages and drawbacks.

Next, a new centroid estimator is proposed, by using the Huber's weight function. This estimator is called the Huber's centroid and it represents a trade-off between the efficiency of the center of mass and the robustness of the median. This estimator is defined starting from the theory of M-estimators and it has one parameter to tune: a threshold that discriminates between outliers and normal data, controlling the estimator's behavior. In addition, an algorithm for the computation of this parameter is given, based on the concept of median absolute deviation that is extended to covariance matrices. Further on, the theoretical part is validated on texture and magnetoencephalography data classification. In the end, the importance of the robust estimators for covariance matrices, introduced in Chapter 3, and that of the robust Huber's estimators are discussed and illustrated in some classification experiments.

Chapter 6 presents an alternative to the classification methods based on global features, proposed in the previous chapters. Sometimes, global descriptors may not be adapted to capture the information contained in signals, or images. For instance, this is the case of non-stationary signals. To address this problem, classification methods based on local descriptors are proposed in this chapter. Therefore, approaches like bag of words, vectors of locally aggregated descriptors, and the Fisher vectors are considered. The first two descriptors have already been generalized for covariance features that live in a Riemannian manifold [Faraki *et al.* 2015b, Faraki *et al.* 2014, Faraki *et al.* 2015a]. Until now, this extension has not yet been possible for the Fisher vectors, due to the lack of some appropriate probabilistic generative models. Based on the Riemannian Gaussian and Laplace distributions presented in Chapter 4, the Riemannian Fisher vectors are defined in this chapter. In the end, their potential is studied for texture image classification.

Chapter 7 synthesizes the main conclusions of this work and it presents some perspectives.

1.3 PhD Context

The work presented in this thesis has been accomplished in the context of a co-tutelle agreement between the University of Bordeaux, France and the Technical University of Cluj-Napoca, Romania, making possible the interweaving of several research tracks, such as texture image classification and directional diffusion based image filtering.

In addition, this thesis has been integrated in an international research project supported by the French Foreign Affairs and International Development Ministry and by the Executive Agency for Higher Education, Research, Development and Innovation Funding Romania, under the projects 32619VL and PNII Capacitati 779/27.06.2014.

Moreover, the thesis has been co-funded by the National Center for Scientific Research (CNRS) and Bordeaux Sciences Agro.

Chapter 2

Textures in Image Processing

Contents

2.1 Textures in Image Analysis

Texture represents an important aspect in the visual perception, involved in the characterization and identification of the objects around us.

In the recent years, this property has been extensively studied in image analysis and several databases containing different texture samples have been created. Two well known examples of texture databases are the VisTex and Outex databases, illustrated in Figure 2.1 and Figure 2.2.

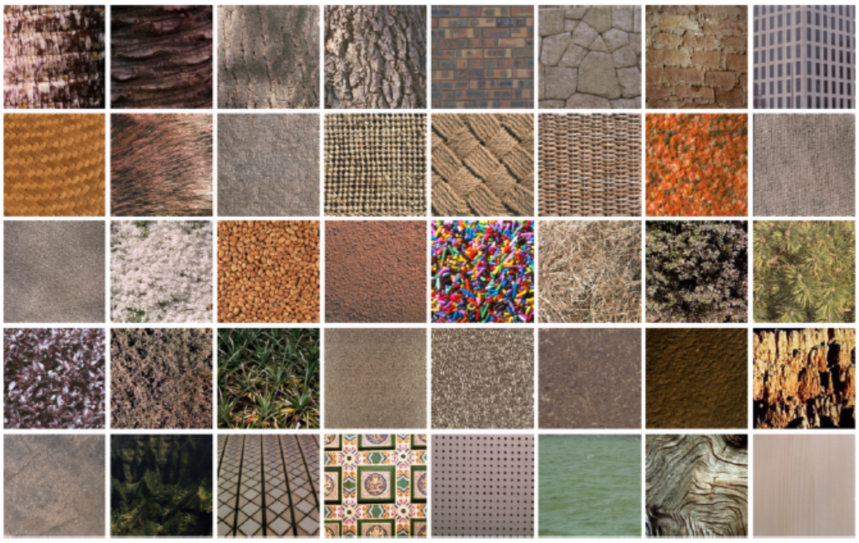


Figure 2.1: VisTex texture database.

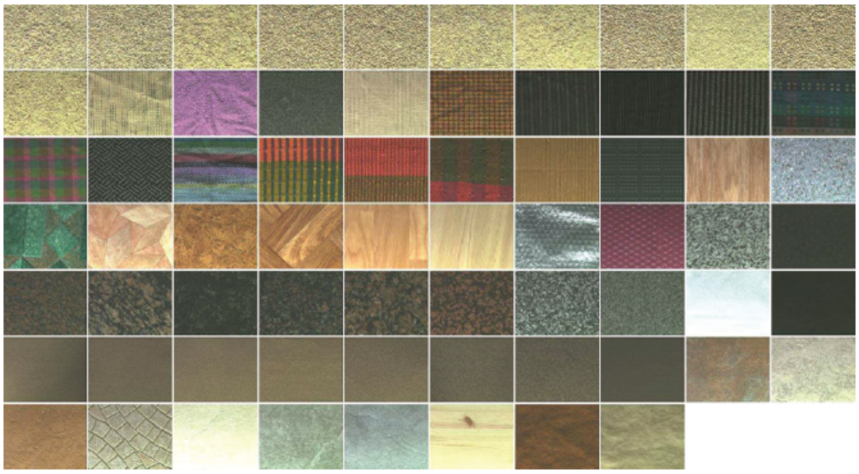


Figure 2.2: Outex_TC000_13 texture database.

From the human point of view, the word "texture" has both a tactile and a visual interpretation. Adjectives like "fine", "coarse", "rough", "smooth", "regular", "irregular", "metallic", "wooden", etc. are frequently used to describe textures, being very hard to translate into some mathematical models. Therefore, many different definitions are available in the computer vision community, trying to capture all the complexity of this concept. In the following, some of these definitions are given.

2.1.1 Definition of Textures

One of the first definitions for textures has been given by Julesz, based on the psychophysics of texture perception [Julesz 1962]. In his work, he claimed that textures can be discriminated by means of first and second order spatial statistics. Later on, he gave some counter-examples for its own theory, by building textures that are different, but with the same second order statistics [Julesz *et al.* 1978]. Thus, he developed another definition, lying on the concept of textons [Julesz 1981, Julesz 1986].

Moreover, according to Haralick, texture is one of the three fundamental pattern elements used in human interpretation of images, along with spectral and contextual features [Haralick *et al.* 1973]. In his opinion, the texture refers to the spatial distribution of gray tones, being an important characteristic of all surfaces.

On the other hand, Tamura has evaluated textures through six properties [Tamura *et al.* 1978]: coarseness, contrast, directionality, line-likeness, regularity and roughness, while Amadasun considered features as busyness, complexity and texture strength [Amadasun & King 1989]. Starting from their works, Rao has identified the smallest set of features that are able to discriminate between textures, that are repetition, orientation and complexity [Rao & Lohse 1993].

In conclusion, even though textures are easily identified and classified by human beings, they do not have a unique definition that can be used in computer vision applications. In order to capture all the wide variety of information lying in textures, different types of descriptors have been proposed in the literature. In the following, some of the employed methods are presented.

2.1.2 Textural Features Extraction

2.1.2.1 Methods Based on Descriptive Statistics

These methods define the texture by means of the spatial distribution of the contained gray values and they include the gray level co-occurrence matrices [Haralick *et al.* 1973], the autocorrelation features [Tuceryan & Jain 1993] and the variograms [Matheron 1963, Curran 1988], the local binary patterns [Ojala *et al.* 1996], etc.

Gray level co-occurrence matrices (GLCM) have been introduced first in [Haralick *et al.* 1973] based on the assumption that for a grayscale image, the textural information lies "in the overall spatial relationship that the gray tones in the image have to one another". This spatial dependence is expressed by means of a matrix containing the relative frequencies of occurrences of two gray tones for two neighboring pixels. In this case, two pixels are neighbors in terms of a predefined distance

d and direction α. Excepting the image bordering pixels, eight nearest neighbors and four angle values are usually considered, as represented in Figure 2.3.

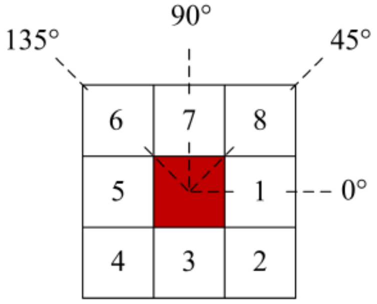


Figure 2.3: The eight angular nearest neighbors of a pixel.

In practice, in order to reduce the size of the GLCM for an image, a quantization step is required first.

Let I be an image of size $W \times H$ and Q the number of quantization levels. In this case, the GLCM will be a $Q \times Q$ matrix. Each element (i, j), $i, j = 0, \ldots, Q-1$ of this matrix represents the number of times gray tones i and j have been neighbors in image I, in terms of distance d and angle α. Mathematically, this can be expressed as:

$$GLCM_{dx,dy}(i,j) = \sum_{x=1}^{W}\sum_{y=1}^{H} \begin{cases} 1, & \text{if } I(x,y) = i \text{ and } I(x+dx, y+dy) = j \\ 0, & \text{otherwise} \end{cases}, \tag{2.1}$$

where dx and dy are the distances according to x and y, i is the gray level of the current pixel and j is the gray level of the neighboring one. To cancel the influence of the image's size, the obtained matrix is normalized by the number of pixels in I.

Further on, starting from the GLCM, a set of 14 textural descriptors can be extracted, expressing image properties like homogeneity and contrast, or measuring the complexity and the nature of gray level transitions [Haralick *et al.* 1973]. In the following, some of these descriptors are detailed, knowing that $P_{i,j}$ denotes the probability of occurrence of neighboring gray levels i and j, that is the element (i, j) of the normalized GLCM:

- *The homogeneity* is given by:

$$\sum_i \sum_j \frac{P_{i,j}}{1+(i-j)^2}. \tag{2.2}$$

  For homogeneous regions, this descriptor will have relative great values, while small values will indicate heterogeneous regions.

- *The entropy* measures the randomness, or the degree of irregularity existing in the image and it is defined as:

$$-\sum_i \sum_j P_{i,j} \ln P_{i,j}. \tag{2.3}$$

Homogeneous regions are characterized by high entropy values, while the heterogeneous, or irregular regions have low entropy.

- *The correlation* measures the linear dependence of gray tones and it is defined as:

$$\sum_i \sum_j P_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}}, \tag{2.4}$$

where μ_i, μ_j σ_i and σ_j are the means and the standard deviations of the marginal distributions associated with each normalized element $P_{i,j}$.

Autocorrelation features are measures of the texture's regularity and coarseness [Tuceryan & Jain 1993]. For instance, coarse textures are characterized by values that slowly drops off. On the other hand, for regular textures, peaks and valleys should be observed in the function's graphical representation.

Variograms are methods used to characterize the spatial dependence between pixels, based on the definition of the semivariogram function, introduced first in [Matheron 1963].

For an image I, the semivariogram function γ is given by:

$$\gamma(d) = \frac{1}{2N} \sum_{i=1}^{N} [I(x_i) - I(x_i + d)]^2, \tag{2.5}$$

where d is the distance between two pixels, N is the number of pixels separated by distance d and $I(x_i)$, respectively $I(x_i+d)$ are the intensities of pixels x_i and x_i+d.

Next, the variogram is obtained as being $2\gamma(d)$ and it measures the dissimilarity between spatially separated pixels [Curran 1988]. More precisely, large values of $\gamma(d)$ indicate less similar pixels. To describe the pixel's correlation, several parameters can be extracted, like the support, lag, sill, range, nugget variance and spatially dependent structural variance.

Local binary patterns (LBP) have been proposed in [Ojala *et al.* 1996] as a particular case of the texture descriptors introduced in [Wang & He 1990]. In order to obtain the LBP for an image I, spatial neighborhoods of 3×3 are extracted for each pixel and a binary sequence is obtained, as follows. First, a comparison is made between the neighborhood's central pixel and each of its 8 neighbors. If the intensity of the central value is smaller than the value of its neighbor, an element equaling 1 is considered for the binary vector, and 0 otherwise. In the end, an 8-digit binary number is obtained, that is usually converted to decimal. By considering all the decimal numbers obtained from the entire image, a histogram is computed, in order to measure the frequency of occurrence of each number. The method has been generalized in [Ojala *et al.* 2002] for different types of neighborhoods and an efficient approach for gray scale and rotation invariant texture classification has been developed.

2.1.2.2 Methods Based on Stochastic Models

These methods imply the characterization of the textural information by using stochastic models. In order to apply them, two steps are needed. First, the textures are analyzed using the multiscale, or the multiresolution representation and then, the obtained coefficients are modeled by means of statistical tools. These two steps are detailed in the following.

1) **Texture analysis**

Multiscale, or multiresolution approaches, have been developed based on the study of human visual perception. The research carried out in this direction has shown that the human brain is capable to perform a multiscale analysis of images [Tamura *et al.* 1978, Landy & Graham 2004]. In this context, the Fourier transform [Georgeson 1979], the Gabor filters [Turner 1986, Jain & Farrokhnia 1991], the wavelet transform [Mallat 1989], the curvelets [Candes & Donoho 1999, Boubchir *et al.* 2010], etc. can be used for capturing the textural information.

Gabor filters have been introduced in [Marĉelja 1980] as models for the simple cells in the visual cortex, showing their importance in image analysis. Mathematically, the following definitions can be formulated.

First, in the spatial domain, the Gabor function $g(x, y)$ is given by a sinusoidal plane wave of frequency f_0 and phase ϕ, modulated by a Gaussian envelope and it is expressed as [Jain & Farrokhnia 1991]:

$$g(x,y) = \exp\left\{ -\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right\}\cos(2\pi f_0 x + \phi), \tag{2.6}$$

where σ_x and σ_y are the standard deviation of the Gaussian envelope along the x and y axis. Moreover, in the frequency domain, considering the phase $\phi = 0$, the function $g(x, y)$ in (2.6) becomes:

$$G(u,v) = A\left(\exp\left\{ -\frac{1}{2}\left(\frac{(u-f_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right)\right\} + \exp\left\{ -\frac{1}{2}\left(\frac{(u+f_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right)\right\}\right), \tag{2.7}$$

where u and v are the horizontal and vertical spatial frequencies, $\sigma_u = 1/(2\pi\sigma_x)$ and $\sigma_v = 1/(2\pi\sigma_y)$ are the corresponding standard deviations and $A = 2\pi\sigma_x\sigma_y$.

Starting from these functions, filter banks can be built and used for image decomposition at different levels and orientations. The obtained images characterize the textural information at these resolutions and orientations and they represent the features used further in applications like image retrieval [Manjunath & Ma 1996], segmentation [Jain & Farrokhnia 1991], texture discrimination [Turner 1986], etc.

Wavelet decomposition has been introduced in [Mallat 1989] and it represents another approach for multiresolution image processing. By using this technique, the image is decomposed in orthogonal and independent subbands, obtained by considering some basis functions, defined as:

$$\psi_{a,b} = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right), \tag{2.8}$$

with a and b being the scale and the location parameters. For an image $f(x, y)$, the wavelet decomposition is performed as:

$$c_{i,j} = \int_{-\infty}^{+\infty} f(x,y)\frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}, \frac{y-b}{a}\right), \tag{2.9}$$

where $c_{i,j}$ are the wavelet coefficients.

In practice, for image decomposition, filter banks of low pass filters (L) and high pass filters (H) are applied along the rows and columns. As a result, four subbands are obtained by combining the two filters. These subbands consist in the image approximation (LL), horizontal (LH), vertical (HL) and diagonal (HH) coefficients. The decomposition is a recursive process and for the next level, the LL subband is used. In addition, a downsampling by a factor of two is considered at each level.

In order to obtain this type of image decomposition, discrete wavelet functions, such as Haar [Haar 1910] and Daubechies [Daubechies 1992] wavelets can be employed.

In the end, the coefficients in each wavelet subband can be described by using statistical models, as shown in Figure 2.4.

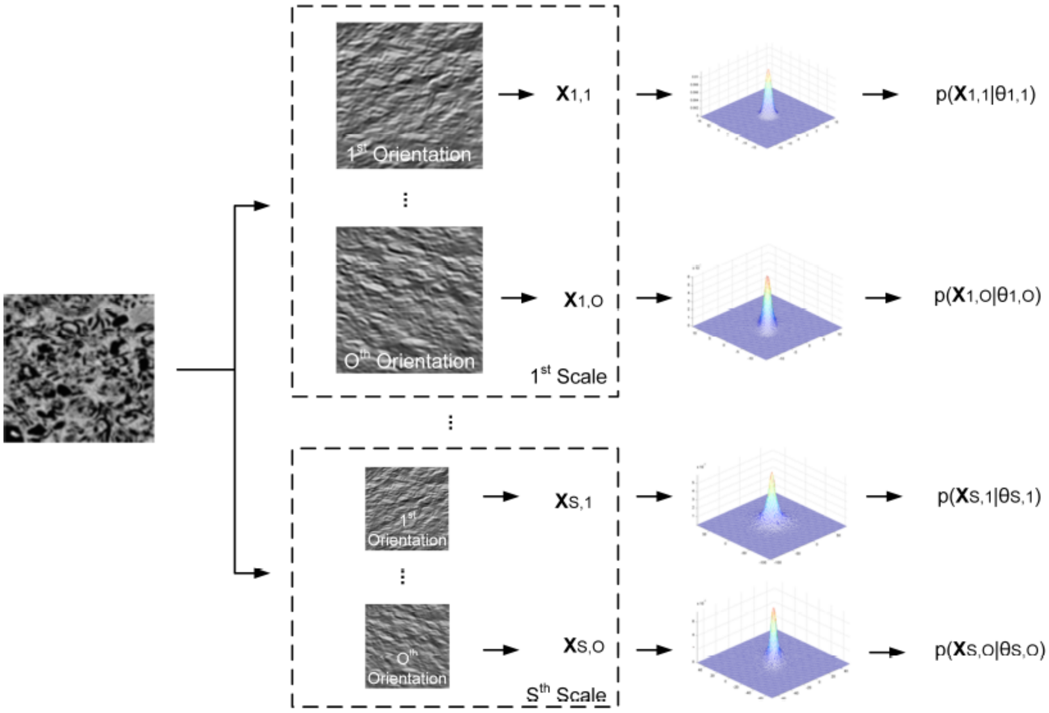


Figure 2.4: Wavelet subbands statistical modeling.

2) **Stochastic modeling**

Once that the texture analysis is accomplished and the textural information is extracted, the filtered elements can be statistically modeled, to obtain the final texture signature. Recently, many statistical models have been proposed. These approaches include the univariate generalized Gaussian distributions [Do & Vetterli 2002], Gamma distributions [Mathiassen *et al.* 2002] and Bessel K forms [Srivastava *et al.* 2002] that will be detailed next.

Generalized Gaussian distributions (GGD) have been proposed in [Do & Vetterli 2002] for modeling the marginal density of the wavelet coefficients in a particular subband. The probability density function describing this model is:

$$p(x|\alpha,\beta) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} \exp\left\{-\left(\frac{|x|}{\alpha}\right)^{\beta}\right\}, \tag{2.10}$$

where $\Gamma(\cdot)$ is the Gamma function, α is the scale parameter and β is the shape parameter. By means of the maximum-likelihood principle, these two parameters can be estimated, giving in the end, the texture's signature. In the same work, the GGD model along with the Kullback-Leibler divergence, as similarity measure, have been successfully used in the context of image retrieval.

Gamma distributions have been considered in [Mathiassen *et al.* 2002] to model features extracted using Gabor filters for texture image classification.

Bessel K forms (BKF) represent another probability model, proposed in [Srivastava *et al.* 2002] for characterizing the output of bandpass filters used in target recognition.

Even though all these univariate models have been successfully used for modeling filtered coefficients, they cannot take into account all the information lying in signals, like the spatial, or spectral dependencies. In order to alleviate this problem, multivariate models have been proposed, including the multivariate Bessel K form distributions [Boubchir *et al.* 2010], copula based distributions [Kwitt *et al.* 2009, Lasmar & Berthoumieu 2014], or the family of multivariate elliptical distributions. This latter, contains the multivariate generalized Gaussian distributions [Verdoolaege & Scheunders 2011], the spherically invariant random vectors [Yao 1973], and multivariate Gaussian distributions, as particular cases.

Multivariate Bessel K form distributions are an extension of the BKFs and they have been introduced in [Boubchir *et al.* 2010] to capture the between-scale and within-scale dependencies between image detail coefficients in wavelet and curvelet domain.

Copula based distributions have been used to model the wavelet coefficients of multichannel images. For instance, in [Kwitt *et al.* 2009] a model based on the two-parameter Weibull distributions and on the multivariate Student-t copula has been developed and applied to texture retrieval, by using the Kullback-Leibler divergence, as similarity measure. Moreover, in [Lasmar & Berthoumieu 2014] Gaussian copula multivariate models have been presented and used for texture image retrieval.

Multivariate generalized Gaussian distributions (MGGD) have been considered in [Verdoolaege & Scheunders 2011, Pascal *et al.* 2013] to describe the wavelet coefficients extracted from multicomponent images, such as color, or multispectral images. By means of MGGDs, the correlation between spectral bands in the wavelet domain has been modeled, showing significant classification improvements with respect to univariate GGDs.

Spherically invariant random vectors (SIRV) have been introduced for the first time in [Yao 1973] and then studied in [Gini & Greco 2002, Pascal *et al.* 2006, Vasile *et al.* 2010], in the context of radar applications. In this case, the observed vector $\mathbf{k}$ is obtained by multiplying the square root of the parameter τ with the complex, circular Gaussian random vector $\mathbf{z}$, of zero mean and covariance matrix $\mathbf{M}$:

$$\mathbf{k} = \sqrt{\tau}\mathbf{z}, \tag{2.11}$$

where τ and z are independent. It yields that the observed vector $\mathbf{k}$ is characterized by the following probability density function:

$$p_{\mathbf{k}}(\mathbf{k}) = \int_0^{\infty} p_{\mathbf{z}}(\mathbf{k}|\tau\mathbf{M})p_{\tau}(\tau)d\tau, \tag{2.12}$$

where $p_{\mathbf{z}}(\,\cdot\,)$ is the probability density function of the multivariate Gaussian distribution.

Multivariate Gaussian distributions represent a particular case of the multivariate models introduced above and they are the density model used further in this thesis.

In our case, the textural information is extracted by means of multiscale approaches. Therefore, each image is filtered using the Daubechies db4 transform. Next, each wavelet subband is statistically modeled by zero-mean multivariate Gaussian distributions (MGDs). This choice has been made based on the fact that this probability density function has desirable properties. More precisely, the geodesic distance, which is the similarity measure considered in this work, has a closed form for the MGD. This is not the case for other distributions, like the SIRV model, for which the geodesic distance can only be approximated. For example, a linear approximation of the geodesics has been considered in [Bombrun *et al.* 2011b] for the multivariate Student-t distribution. On the other hand, the adaptation of the proposed approach to more complex models is taken into consideration for future work.

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of N independent and identically distributed random vectors of dimension m, issued from a zero-mean multivariate Gaussian distribution. The probability density function describing the set is the following:

$$p(\mathbf{x}|\mathbf{M}) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{M}|}} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{M}^{-1} \mathbf{x}\right). \tag{2.13}$$

For this model, the parameter vector $\boldsymbol{\theta}$ represented in Figure 2.4 is the covariance matrix $\mathbf{M}$, which gives the final texture signature.

In the following, the importance of covariance matrices in signal and image processing will be detailed.

2.2 Covariance Matrices as Signal and Image Descriptors

2.2.1 Importance in Image Analysis

Covariance matrices are used in a wide variety of applications in signal and image processing, including array processing [Ollila & Koivunen 2003], radar detection [Greco *et al.* 2014,Chen *et al.* 2011,Yang *et al.* 2010,Barbaresco *et al.* 2013,Mahot *et al.* 2013], medical image segmentation [de Luis-García *et al.* 2011], face detection [Robinson 2005], vehicle detection [Mader & Reese 2012], etc. Another research direction concerns the signal and image classification, where covariance matrices can be used to model different kinds of dependence, like spatial, temporal, spectral, polarimetric dependence, etc [Formont *et al.* 2011, Barachant *et al.* 2013, Said *et al.* 2015a, Faraki *et al.* 2015a].

Being elements in the space $\mathcal{P}_m$ of $m \times m$ real symmetric and positive definite matrices, several distributions have been proposed to model them, such as the Wishart distribution [Wishart 1928], those issued from the so-called product model [Lee *et al.* 1993,Freitas *et al.* 2003,Bombrun & Beaulieu 2008,Bombrun *et al.* 2011a] and those inspired from a geometric point of view: the Riemannian distributions [Said *et al.* 2015b,Hajri *et al.* 2016], etc.

2.2.2 Statistical Models for the Space of Covariance Matrices

Wishart distribution (WD) has been introduced in [Wishart 1928] and it represents the multidimensional version of the χ^2 distribution.

Let $\mathbf{M}$ be an $m \times m$ symmetric and positive definite matrix having a Wishart distribution with n degrees of freedom. The probability density function characterizing it is defined as:

$$p(\mathbf{M}|n, \mathbf{S}) = \frac{|\mathbf{M}|^{\frac{n-m-1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}^{-1}\mathbf{M}\right)\right\}}{2^{\frac{nm}{2}} |\mathbf{S}|^{\frac{N}{2}} \Gamma_m\left(\frac{n}{2}\right)}, \tag{2.14}$$

where $|\cdot|$ denotes the matrix determinant, $\Gamma_m(\cdot)$ is the multivariate Gamma function, $\mathbf{S}$ is the scale matrix and $n \geq m$.

This distribution has been used in applications like motion retrieval [Saint-Jean & Nielsen 2013], or to model complex-valued data, in the context of polarimetric image classification [Lee *et al.* 1999]. On the other hand, this model is no longer appropriate for images which do not fulfill the assumption of homogeneity, such as the high resolution PolSAR images. A solution to this problem is to use the product model based distributions, that are detailed next.

Distributions issued from the scalar product models consist in expressing the observed covariance matrix $\mathbf{M}$, of textured regions, as being obtained by multiplying the scalar parameter τ by the scatter matrix $\boldsymbol{\Sigma}$:

$$\mathbf{M} = \tau\boldsymbol{\Sigma}, \tag{2.15}$$

where τ and $\boldsymbol{\Sigma}$ are independent and the scatter matrix $\boldsymbol{\Sigma}$ follows a complex Wishart distribution. To be identifiable, a normalization constraint should be imposed on the model. In practice, the trace of the scatter matrix Σ is generally imposed to be equal to m. Some others normalizations can be considered, such as imposing a condition on the determinant of the scatter matrix, or imposing that the mean of τ is equal to 1. Moreover, the probability density function of the covariance matrix $\mathbf{M}$ is given by:

$$p_{\mathbf{M}}(\mathbf{M}) = \int_0^\infty p_{\boldsymbol{\Sigma}}(\mathbf{M}|\tau\boldsymbol{\Sigma}) p_\tau(\tau) d\tau. \tag{2.16}$$

Depending on the choice of $p_\tau(\tau)$, different models can be obtained, such as the $\mathcal{K}$ [Lee *et al.* 1993], $\mathcal{G}^0$ [Freitas *et al.* 2003], KummerU [Bombrun & Beaulieu 2008], $\mathcal{M}$ and $\mathcal{W}$ [Bombrun *et al.* 2011a] distributions. For these models, τ follows respectively the Gamma, Inverse Gamma, Fisher, Beta and Inverse Beta distributions.

Even though these models can be efficiently used, they do not take into consideration the intrinsic geometry of the data. In order to address this problem, a new class of distributions have been proposed in the literature, that are introduced in the following.

Riemannian distributions have been recently proposed to model the within-class variability of images. First, inspired by the conventional multivariate Gaussian distribution, the Riemannian Gaussian distribution (RGD) has been introduced in [Said *et al.* 2015b]. In this case, the probability density function is given by:

$$p\left(\mathbf{M}|\bar{\mathbf{M}}, \sigma\right) = \exp\left\{-\frac{d^2(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2}\right\}, \tag{2.17}$$

where $d(\cdot)$ is not the classical Mahalanobis distance, but the intrinsic distance on the manifold of covariance matrices. This probability density function is characterized by two parameters, its central element $\bar{\mathbf{M}}$ and its dispersion σ around this central element. For this model, the maximum likelihood estimator (MLE) of the central value corresponds to the Riemannian center of mass. While being efficient to model the mean element, this latter is easily influenced by the presence of aberrant data [Bishop 2007, Afsari 2011, Formont *et al.* 2013]. To overcome this problem, we have introduced a generative model for which the MLE of the central element is the Riemannian median, called the Riemannian Laplace distribution (RLD) [Hajri *et al.* 2016]. Both RGD and RLD are detailed in Chapter 4 and used for signal and texture image classification.

2.3 Conclusions

In this chapter, state-of-the-art methods concerning the extraction of the textural information and the covariance matrix modeling have been presented.

First, classical textural features based on descriptive statistics have been briefly presented. These descriptors include the gray level co-occurrence matrices, the autocorrelation features, the variograms, or the local binary patterns.

Second, feature extraction methods based on statistical modeling have been described. These approaches imply the texture analysis at different scales and orientations and the characterization of the obtain information by stochastic modeling. In this context, the Gabor filters and the wavelet decomposition have been detailed, along with some univariate and multivariate statistical models.

Further on, the methods used in this thesis have been introduced. More precisely, in this work, the textural information is captured by means of the wavelet decomposition and the extracted coefficients are modeled by zero-mean multivariate Gaussian distributions. This process has been illustrated by a general diagram shown in Figure 2.4. Moreover, the parameter of the considered distribution is the covariance matrix. This matrix is an element in the space $\mathcal{P}_m$ of $m \times m$ real symmetric and positive definite matrices which motivates the need of appropriate modeling distributions. In order to respect the geometry of this space, the Riemannian distributions have been chosen for this thesis.

In the following chapters, the classification workflow introduced in Figure 1.1 is resumed and each block of this diagram is detailed in a distinct part.

Chapter 3

Robust Classification Workflow on the Space of Covariance Matrices

Contents

3.1 Introduction

In the classification context, the image processing workflow consists in two steps: feature extraction, and classification.

Various kinds of features can be used for image processing, such as texture, spectral information, polarimetric dependence, etc. One strategy to obtain relevant features is the multiscale image decomposition. This approach has been found to be successful for many image processing applications including filtering [Donoho 1995], segmentation [Aujol *et al.* 2003], or classification [Do & Vetterli 2002].

First of all, during the feature extraction stage, the image is decomposed into a set of wavelet subbands, each of them being modeled by a probability density function with a specific parameter vector. For each subband, the estimated parameter vector composes the signature of the image. Then, during the classification stage, a similarity measure based on a probabilistic metric is computed between the signature vectors.

Simple but effective methods have been proposed to characterize wavelet detail statistics based on univariate models, such as the generalized Gaussian distribution [Do & Vetterli 2002]. Nonetheless, they do not take into account the dependencies existing in the image. To overcome this difficulty, multivariate distributions, including elliptical models [Bombrun *et al.* 2011b, Verdoolaege & Scheunders 2012] and copula based approaches [Kwitt & Uhl 2010, Stitou *et al.* 2009], have been proposed to model the spatial and spectral dependencies in the images. Once the feature vectors are computed for each texture image, a distance (or at least a divergence) is calculated in order to measure the degree of similarity between two images. A well-known choice is the Kullback-Leibler (KL) divergence [Kullback & Leibler 1951], or its symmetric version: the Jeffreys divergence [Jeffreys 1946]. Recently, some authors have proposed to consider the geodesic distance (GD), which has shown superior retrieval rate, compared to the KL divergence [Verdoolaege & Scheunders 2012].

Starting from this general framework, the purpose of this chapter is to introduce a robust classification workflow. For that, the concept of robustness is addressed at different levels. First, it can be considered during the modeling step when covariance matrices are estimated for each image. Second, the robustness can be investigated during the decision rule of the classifier. Third, for the proposed application of Polarimetric Synthetic Aperture Radar (PolSAR) image classification, a preprocessing filtering step consisting in speckle reduction is introduced to reduce the influence of outliers in the estimation and classification performances. The proposed workflow is illustrated in Figure 3.1, where the colored blocks show the steps where the proposed classification method brings some changes. The main contributions of this chapter are detailed hereafter.

As mentioned earlier, in Chapter 2, the wavelet coefficients are modeled in this thesis by using the zero-mean multivariate Gaussian distribution (MGD). This probability density function is characterized by the covariance matrix, which represents the final data signature. In order to estimate this parameter, different methods are

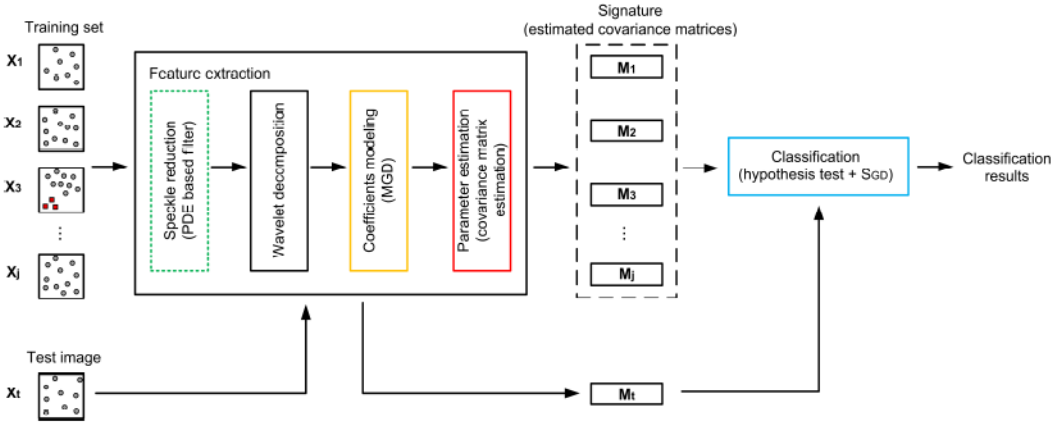

Training set
X1
X2
X3
Xj
Fcaturc cxtraction
Speckle reduction (PDE based filter)
Wavelet deccmposition
Coefficients modeling (MGD)
Parameter estimation (covariance matrix estimation)
Signature (estimated covariance matrices)
M1
M2
M3
Mj
Classification (hypothesis test + SGD)
Classification results
Test image
Xt
Mt

Figure 3.1: Classification workflow for robust image classification on the space of covariance matrices.

proposed in the literature, like the sample covariance matrix (SCM), the fixed point estimator (FP), also known as the Tyler's estimator [Tyler 1987], the class of M-estimators [Huber 1964, Tyler 1987], etc. At this step, robust estimation algorithms are needed to deal with the possible existing noise or image artifacts. An example is presented in Figure 3.1, where $\mathbf{X} = \{X_1, \ldots, X_j\}$ is a set of observations. For each observation, the data points are represented by gray circles, while the outliers are identified by red squares. In this case, the robust covariance matrix estimator has to be able to reduce the impact of outliers in the estimation process. As a result, the robust image signature is obtained, which will be used during the classification process.

In a classification or texture retrieval experiment, a nearest neighbor classifier is frequently considered. In such case, a test image, denoted by X_t in Figure 3.1, is labeled to the class of the closest training image, but nothing tells that the test image is well classified, especially for noisy datasets. A hypothesis test should be performed to regulate the false alarm rate. Inspired from previous works on the KL divergence [Kupperman 1957] and on the family of (h, ϕ) divergences [Salicru *et al.* 1994, Nascimento *et al.* 2010], a new statistical hypothesis test based on the geodesic distance is introduced in this chapter [Ilea *et al.* 2015b, Ilea *et al.* 2015c, Ilea *et al.* 2015a]. The advantage of using the geodesic distance lies in its property of being a distance measure, which is symmetric and respects the triangle inequality.

The main application of the robust classification approaches introduced in this chapter is represented by the classification of PolSAR data. These images are characterized by a multiplicative noise, called speckle, which makes difficult their analysis. Therefore, in order to improve the algorithm's robustness, a preprocessing step is added, consisting in the speckle reduction. To this aim, a directional diffusion filter is proposed [Terebes *et al.* 2015, Terebes *et al.* 2016], based on the partial differential equation formalism. Some experiments are carried out to evaluate the influence of the filtering step on the classification accuracy.

The chapter is structured as follows. Section 3.2 details the covariance matrix estimation process, by analyzing some of the state-of-the-art estimators. Section 3.3 introduces the proposed statistical hypothesis test based on the geodesic distance. First, the test is defined in a general context and then, it is applied to zero-mean multivariate Gaussian distributions MGD for the SCM covariance matrix estimator. Further on, it is used for the case of the robust FP estimator. In addition, its performance is analyzed in terms of efficiency and noise robustness on simulated data. Some comparisons with the SCM estimator are also carried out. Section 3.4 introduces an application for the classification of maritime pine forests, based on simulated and real PolSAR images. Section 3.5 introduces the directional diffusion based filtering used for PolSAR image denosing and the influence of the filtering step on the classification performance is evaluated. Conclusions and future work are finally reported in Section 3.6.

3.2 Covariance Matrices and Estimation Methods

Zero-mean MGDs are characterized by their covariance matrix. In the context of parametric classification methods, this matrix needs to be estimated. In order to obtain robust classification algorithms, robust estimators are desired. In the following, state-of-the-art covariance matrix estimators are presented for a set $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_1}\}$ of N independent and identically distributed random variables (vectors) $\mathbf{x}$ according to an MGD.

3.2.1 Sample Covariance Matrix

The sample covariance matrix (SCM), or the empirical covariance matrix, is one of the most common estimators as it represents the solution of the maximum likelihood (ML) estimator for zero-mean Gaussian distribution. In this case, the SCM estimator $\hat{\mathbf{M}}_{SCM}$ of the covariance matrix $\mathbf{M}$, characterizing $\mathbf{X}$, is given by the following equation:

$$\hat{\mathbf{M}}_{SCM} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T, \tag{3.1}$$

where $(\cdot)^T$ denotes the transpose operator. In [Anderson 1984] the properties of this estimator have been analyzed and it has been shown that SCM is an unbiased estimate, but it has a major drawback: it is not robust to outliers.

3.2.2 Normalized Sample Covariance Matrix

The normalized sample covariance matrix (NSCM) may represent a solution to the non robustness problem. In this case, the estimated covariance matrix $\hat{\mathbf{M}}_{NSCM}$ is given by:

$$\hat{\mathbf{M}}_{NSCM} = \frac{m}{N}\sum_{i=1}^{N} \frac{\mathbf{x}_i\mathbf{x}_i^T}{\mathbf{x}_i^T\mathbf{x}_i}, \tag{3.2}$$

where m is the dimension of vectors $\mathbf{x}$. However, this estimator has two major drawbacks: it is biased and nonconsistent [Pascal *et al.* 2008].

3.2.3 Fixed Point Estimator

The fixed point estimator (FP), also known as the Tyler's estimator [Tyler 1987], is another possible choice to solve the non robustness problem. In this case, the estimated covariance matrix $\hat{\mathbf{M}}$ is obtained by means of a recursive algorithms as the solution of [Gini & Greco 2002, Pascal *et al.* 2008]:

$$\hat{\mathbf{M}}_{it+1} = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{x}_i \mathbf{x}_i^T}{\mathbf{x}_i^T \hat{\mathbf{M}}_{it}^{-1} \mathbf{x}_i}, \tag{3.3}$$

with it being the iteration.

In practice, this recursive algorithm can be initialized with the identity matrix and it converges in about 10 iterations [Conte *et al.* 2002, Pascal *et al.* 2008, Vasile *et al.* 2010].

The FP estimator has a unique solution $\hat{\mathbf{M}}$ up to a scale factor. For any positive scalar $c \neq 0$, if $\hat{\mathbf{M}}$ is a solution of (3.3), then $c\hat{\mathbf{M}}$ is also a solution of (3.3). In the following, the covariance matrix is normalized such that:

$$\mathrm{tr}(\hat{\mathbf{M}}) = m, \tag{3.4}$$

where $\mathrm{tr}(\cdot)$ is the trace operator and m is the vector's dimension. This FP estimator can be interpreted as the ML estimate of the normalized covariance matrix for a Gaussian scale mixture model, where the multipliers τ_i are assumed to be unknown deterministic parameters [Gini & Greco 2002]. Let us recall that a Gaussian scale mixture model admits the stochastic representation $\mathbf{x} = \sqrt{\tau}\mathbf{z}$ where τ is a scalar random variable called multiplier ($\tau \in \mathbb{R}^{+}$) and $\mathbf{z}$ is an independent Gaussian random vector with zero-mean and covariance matrix $\mathbf{M}$.

In [Pascal *et al.* 2008], the properties of FP estimator have been analyzed. In particular, the FP provides a unique solution of (3.3) and it is an unbiased and consistent estimate. Moreover, the FP estimate follows asymptotically a Wishart distribution behavior, with $N\frac{m}{m+1}$ degrees of freedom.

3.2.4 Robust M-estimators

The family of M-estimators covers some other possible robust covariance matrix estimators.

3.2.4.1 Definition

The M-estimators have been introduced in the context of robust theory to tackle the presence of outliers in the dataset or errors in the model. For zero-mean observations, the M-estimator of the covariance matrix is defined as the solution of [Huber 1964, Tyler 1987]:

$$\hat{\mathbf{M}} = \frac{1}{N} \sum_{i=1}^{N} u(\mathbf{x}_i^T \hat{\mathbf{M}}^{-1} \mathbf{x}_i)\ \mathbf{x}_i \mathbf{x}_i^T. \tag{3.5}$$

where $u(\cdot)$ is a positive-valued function, which gives a weight to each observation $\mathbf{x}_i$ in the computation of the covariance matrix. Obviously the weight function $u(\cdot)$ should decrease to zero to ensure that outliers have a smaller contribution to the covariance matrix estimation than other observations. In addition, the weight function $u(\cdot)$ has to fulfill the conditions expressed in [Maronna 1976]:

1. $u(t)$ is non-negative, non-increasing and continuous for $t \geq 0$;
2. $\psi(t) = tu(t)$ is bounded and $K = \sup_{t \geq 0} \psi(t)$;
3. $\psi(t)$ is non-decreasing, and strictly increasing in the interval where $\psi(t) < K$;
4. there exists $a > 0$ such that for every hyperplane H, $p(H) \leq 1 - \frac{1}{K} - a$, where $p(\cdot)$ is the dataset's empirical distribution.

These conditions have been analyzed for the real case in [Maronna 1976] and generalized to complex-valued data in [Ollila & Koivunen 2003].

The family of M-estimators has been extensively studied and it has been found that it is a generalization of covariance matrix ML estimates for the family of elliptical distributions.

Depending on the weight function, various covariance matrix estimators can be defined. For example, if $u(t) = 1$, all the observations have the same weight, resulting in the sample covariance matrix (SCM) estimator. Moreover, if $u(t) = 1/t$, the fixed point (FP) estimator [Tyler 1987], is obtained. It has to be mentioned that even though the SCM and the FP estimators have expressions similar to (3.5), they do not belong to the family of M-estimators, because they do not satisfy some of the conditions defined in [Maronna 1976]. More precisely:

- for the SCM: the upper limit of $\psi(t) = tu(t)$ is infinite;
- for the FP: the weight function $u(\cdot)$ is not defined when $t = 0$.

More interesting, the Huber's estimator [Huber 1964] offers a trade-off between the SCM and the FP. This estimator will be detailed in the following.

3.2.4.2 Huber's Estimator

A possible choice for the weight function $u(\cdot)$ in (3.5) is:

$$u(t) = \min\left(1, \frac{T}{t}\right), \tag{3.6}$$

which gives the Huber's estimator [Huber 1964]. In this expression, T is a predefined threshold value that controls the influence of outliers. If the quadratic term $t = \mathbf{x}_i^T \hat{\mathbf{M}}^{-1} \mathbf{x}_i$ is smaller than T, the Huber's function $u(t)$ is constant, otherwise $u(t)$ will start to decrease. Tuning the parameter T allows to adjust the behavior of the Huber estimator between the SCM and the FP estimator. The properties of this estimator have been analyzed in the literature and used in the context of array processing [Ollila & Koivunen 2003], or the estimation of the directions of arrival [Mahot *et al.* 2013].

3.3 Hypothesis Test for Robust Classification

The hypothesis test represents a decision-making strategy founded on the statistical significance of a result. More precisely, a result is considered to be significant if the probability to obtain it by chance is small with respect to a predefined threshold value.

In practice, the hypothesis tests are used for testing a statement about a population, based on some data measured from a sample [Moon & Stirling 2000]. For instance, in the case of a binary problem, two statements are defined: the *null hypothesis* H_0 and the *alternative hypothesis* H_1. The two hypotheses are disjoint, H_1 being the negation of H_0. The test's goal is to reject H_0, meaning that H_1 is supposed to be true. In order to implement such a test, several steps are needed [Ruch 2012]:

- definition of the null hypothesis H_0 that has to be rejected;
- definition of a discriminant measure between H_0 and H_1, called *statistic*;
- definition of a probability distribution for the considered statistic under H_0;
- definition of a threshold value for the statistic's probability distribution, called *significance level* and denoted by α;
- computation of the sample's statistic value and p-value;
- making the decision.

Next, these steps will be followed to propose a robust statistical test for classification, on the space of covariance matrices. The purpose of this test is to regulate the false alarm rate during the decision making stage.

3.3.1 Definition

Let $\mathbf{X}_1 = \{\mathbf{x}_1^1, \ldots, \mathbf{x}_{N_1}^1\}$ and $\mathbf{X}_2 = \{\mathbf{x}_1^2, \ldots, \mathbf{x}_{N_2}^2\}$ be two sets of N_1 and N_2 independent and identically distributed random variables (vectors) $\mathbf{x}$ according to the parametric models $p(\mathbf{x}|\theta_1)$ and $p(\mathbf{x}|\theta_2)$. Let $\hat{\theta}_1$ and $\hat{\theta}_2$ be the maximum likelihood (ML) estimates computed on these sets. In a classification problem, the aim is to determine if $\mathbf{X}_1$ and $\mathbf{X}_2$ are issued from the same parametric model. Therefore, let consider the following hypothesis test:

$$\begin{cases} H_0 : \theta_1 = \theta_2; \\ H_1 : \theta_1 \neq \theta_2, \end{cases} \tag{3.7}$$

where, the hypothesis H_0 states that $\mathbf{X}_1$ and $\mathbf{X}_2$ are elements of the same class, if their parameters are identical.

Considering the regularity conditions discussed in [Salicru *et al.* 1994], it has been proved in [Kupperman 1957, Salicru *et al.* 1994, Nascimento *et al.* 2010] that under the null hypothesis H_0 and for sample sizes $N_1, N_2 \longrightarrow \infty$, the test statistic S, defined further, follows a chi-square distribution:

$$S(\widehat{\theta}_1, \widehat{\theta}_2) = \frac{2vN_1N_2}{N_1 + N_2}\delta(\widehat{\theta}_1, \widehat{\theta}_2) \xrightarrow[N_1,N_2\to\infty]{} \chi^2_{DF}, \tag{3.8}$$

where the degree of freedom DF is equal to the dimension of the parameter space. In addition, v is a constant depending on the considered similarity measure $\delta(\,\cdot\,)$. For instance, $v = 1$ for the KL divergence [Nascimento *et al.* 2010]. This hypothesis test has been first introduced in [Kupperman 1957] for $\delta(\,\cdot\,)$ being the KL divergence and further generalized in [Salicru *et al.* 1994] for the class of (h, ϕ) divergences. In this chapter, this test is extended to the Rao's geodesic distance, which is the shortest path in the parametric manifold. Indeed, under the null hypothesis H_0, distributions are lying infinitesimally close on the probabilistic manifold and in such case the KL divergence equals half of the squared geodesic distance (GD). Hence, when $\theta_1 = \theta_2$, the test statistic becomes:

$$S_{GD}(\widehat{\theta}_1, \widehat{\theta}_2) = \frac{N_1N_2}{N_1 + N_2}d^2(\widehat{\theta}_1, \widehat{\theta}_2), \tag{3.9}$$

with $d(\,\cdot\,)$ being the geodesic distance and it is asymptotically chi-square distributed with DF degrees of freedom for sufficiently large values of N_1 and N_2. Note that under H_0, the distribution of the statistic S_{GD} is independent of θ_1 and θ_2.

3.3.2 Application to Zero-Mean Multivariate Gaussian Distributions

In the following, $\mathbf{X}_1$ and $\mathbf{X}_2$ are issued from two independent zero-mean multivariate Gaussian distributions (MGDs) having the parameter vectors represented by the covariance matrices $\mathbf{M}_1$ and $\mathbf{M}_2$. The probability density function describing these sets has been introduced in Chapter 2 and it is recalled in the following:

$$p(\mathbf{x}|\mathbf{M}) = \frac{1}{\sqrt{(2\pi)^m|\mathbf{M}|}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{M}^{-1}\mathbf{x}\right), \tag{3.10}$$

where m is the vector's $\mathbf{x}$ dimension and $(\,\cdot\,)^T$ is the transpose operator.

Considering zero-mean MGDs, the earlier mentioned similarity measures between the two estimated covariance matrices $\hat{\mathbf{M}}_1$ and $\hat{\mathbf{M}}_2$ have the following definitions:

- Kullback-Leibler divergence [Kullback & Leibler 1951]:

$$KL(\hat{\mathbf{M}}_1, \hat{\mathbf{M}}_2) = \frac{1}{2}\left[\mathrm{tr}(\hat{\mathbf{M}}_2^{-1}\hat{\mathbf{M}}_1) - m - \ln\frac{|\hat{\mathbf{M}}_1|}{|\hat{\mathbf{M}}_2|}\right], \tag{3.11}$$

  where m is the dimension of the vector space, and $\mathrm{tr}(\cdot)$ is the trace operator;

- Rao's geodesic distance [James 1973]:

$$d(\hat{\mathbf{M}}_1, \hat{\mathbf{M}}_2) = \left[\frac{1}{2}\sum_i (\ln\lambda_i)^2\right]^{\frac{1}{2}}, \tag{3.12}$$

  where $\hat{\mathbf{M}}_1$ and $\hat{\mathbf{M}}_2$ are the SCM estimators of the covariance matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, λ_i, $i = 1\dots m$ are the eigenvalues of $\hat{\mathbf{M}}_2^{-1}\hat{\mathbf{M}}_1$ and m is the size of covariance matrices. More details on this similarity measures can be found in Section 4.2.2.

In this case, the null hypothesis $\mathbf{M}_1 = \mathbf{M}_2$ can be rejected at a level α if:

$$Pr(\chi^2_{DF} > S_{GD}(\hat{\mathbf{M}}_1, \hat{\mathbf{M}}_2)) \leq \alpha. \tag{3.13}$$

Here, knowing that real-valued covariance matrices are considered, the degree of freedom is:

$$DF = \frac{m(m+1)}{2}, \tag{3.14}$$

where m is the dimension of the covariance matrix. The rejection of H_0 is illustrated in Figure 3.2.a.

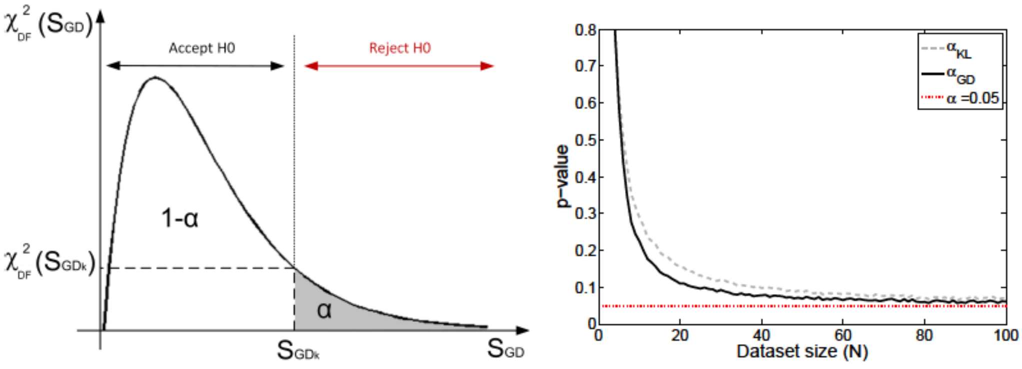
Further on, some simulation results are displayed to evaluate the potential of the proposed statistical hypothesis test on a simulated dataset. The sets $\mathbf{X}_1$ and $\mathbf{X}_2$ are generated as N_1 and N_2 independent and identically distributed random vectors distributed according to a zero-mean MGD, having the covariance matrix of the form:

$$\mathbf{M}(i,j) = \rho^{|i-j|} \text{ for } i,j \in [\![1, m]\!]. \tag{3.15}$$

For each set $\mathbf{X}_1$ and $\mathbf{X}_2$, the covariance matrix is estimated according to the maximum likelihood principle by using the SCM estimates. The significance level α is set to 0.05 and 10^4 Monte Carlo iterations are considered. Figure 3.2.b draws the evolution of the estimated p-value as a function of the dataset size ($N_1 = N_2 = N$ in this experiment) for the SCM estimate with $\rho = 0.5$.

In this figure, the solid line corresponds to the geodesic distance, while the dashed line corresponds to the Kullback-Leibler divergence. As expected, the estimated p-value converges to the significance level α for sufficiently large N. The simulation results have shown that the dataset should contain at least 50 observations to ensure that the statistic follows a chi-squared distribution under the null hypothesis H_0, when using the SCM estimate. In addition, it can be observed that the convergence is faster for the geodesic distance than for the Kullback-Leibler divergence. In the following, only the geodesic distance will be considered.

In the next part, this hypothesis test is used in the case of robust covariance matrix estimators, more precisely, in the case of the fixed point estimator.



(a) (b)

Figure 3.2: (a) Rejection of the null hypothesis and (b) convergence of the estimated p-value as a function of the dataset size, when the SCM estimator is used (Source: [Ilea *et al.* 2015b] ©[2015] IEEE).

3.3.3 Application to Robust Estimators

3.3.3.1 Classification Efficiency

The proposed hypothesis test based on the statistic S_{GD} is used further for implementing a two-class classification algorithm for simulated data. The experiment consists in defining two independent and identically distributed random sets of vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ of size N_1 and N_2 distributed according to two MGDs and having the covariance matrices $\mathbf{M}_1$ and $\mathbf{M}_2$. A third dataset $\mathbf{X}_t$ of size N_t and covariance matrix $\mathbf{M}_t$ has been defined in the same manner. The objective of the implemented algorithm is to classify $\mathbf{X}_t$ in one of the two available groups, by choosing the one with the most similar covariance matrix. In this experiment, it is considered that $\mathbf{X}_t$ should be of class 2, by generating it using the same parameters as for $\mathbf{X}_2$. Under these assumptions, the hypothesis test consists in verifying if the distribution of $\mathbf{X}_t$ has the same parameter vector as the one of $\mathbf{X}_2$, or in other words, if $\mathbf{X}_t$ is of class 2.

Figure 3.3 presents the influence of the estimation algorithm and the influence of datasets' size on the classification performance. The simulations are carried out for a 3-dimensional dataset ($m = 3$) with $N_1 = 100$, 1000 and 10000 and $N_2 = N_t = 1000$. Several values are tested for the covariance matrix $\mathbf{M}$ (ρ_2 and ρ_t ranging from 0.1 to 0.7, while ρ_1 is fixed to 0.1). Each time, $\mathbf{M}$ is estimated by the SCM (dashed lines) and FP (solid lines) algorithms. 10^4 Monte Carlo iterations are performed to compute average performances.

As observed, the best performances are obtained for the SCM estimate compared to the FP one, illustrating the efficiency of this former. This observation is natural since the experiment has been carried out in a purely Gaussian context. The next experiment is designed in order to analyze the robustness by considering some noisy data.

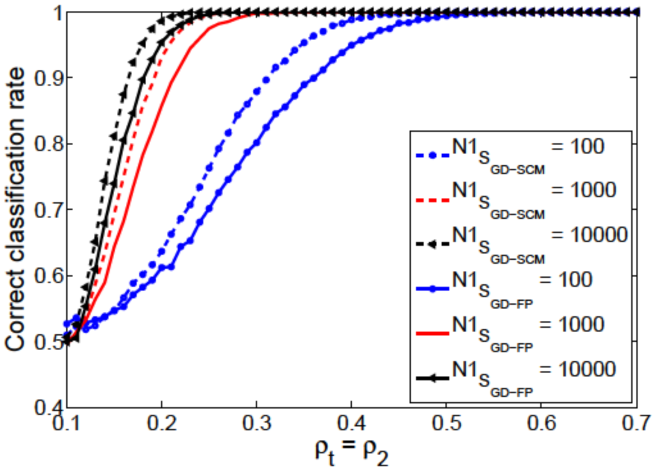


Figure 3.3: Correct classification rate of $\mathbf{X}_t$ in class 2 by using the S_{GD} with the SCM and FP estimates, if $N_1 = 100$, 1000, 10000, and $N_2 = N_t = 1000$ (Source: [Ilea *et al.* 2015b] © [2015] IEEE).

3.3.3.2 Noise Robustness

The performances of the SCM and FP estimates are now compared in terms of noise robustness for the statistic S_{GD}. Thus, two datasets $\mathbf{X}_1$ and $\mathbf{X}_2$ are generated as independent and identically distributed random vectors distributed according to a zero-mean MGD of covariance matrix $\mathbf{M}$. The set $\mathbf{X}_2$ is next corrupted by an independent additive white Gaussian noise of covariance matrix $\sigma^2\mathbf{I}_m$, σ^2 being the noise variance and $\mathbf{I}_m$ is the identity matrix.

The significance level α is set to 0.05 and several values are tested for the covariance matrix $\mathbf{M}$ ($\rho = 0.25$, 0.5, and 0.75). 10^3 Monte Carlo iterations are considered to estimate the classification rate and the results are displayed in Figure. 3.4.

The dashed and solid lines correspond to the SCM and FP estimates. Clearly, the FP estimator is much more robust than the SCM, especially for smaller values of ρ.

In this chapter, a new statistical hypothesis test for robust image classification has been introduced. First, the proposed statistical hypothesis test has been defined, based on the geodesic distance. Next, its properties have been analyzed in the case of the zero-mean multivariate Gaussian distribution, by studying its asymptotic distribution under the null hypothesis H_0. In the end, the performance of the proposed classifier has been addressed by analyzing its noise robustness and comparisons have been made for the SCM and FP estimators. Further on, the statistic S_{GD}, involved in the definition of this test, will be applied to PolSAR image classification.

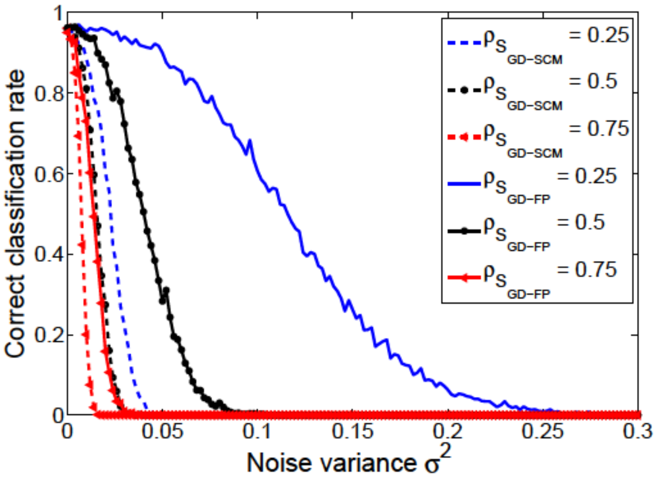


Figure 3.4: Evolution of the performances of S_{GD} as a function of the noise variance σ^2, by using the SCM and FP estimate for various covariance matrices: $\rho = 0.25$, 0.5, 0.75 and $m = 3$ (Source: [Ilea *et al.* 2015b] © [2015] IEEE).

3.4 Application to PolSAR Image Classification

The experiments, carried out in this section, have several purposes:

- First, the proposed similarity measure S_{GD} is applied to simulated and real PolSAR image classification. In order to obtain the simulated dataset, a PolSAR scenes simulator [Williams 2006] is used.

- Second, the influence of the acquisition parameters (incidence angle, spatial resolution, number of polarimetric channels) on classification accuracy is analyzed by considering several simulated databases.

- Third, different strategies are proposed in order to model the dependencies (spatial, polarimetric) present in PolSAR images.

3.4.1 Database

3.4.1.1 Simulated L-band PolSAR Images

The simulated dataset is created by using the PolSARproSim software [Williams 2006]. This software provides fully polarized simulated SAR images of forest, displaying properties consistent with real SAR imagery [Williams 2006]. Images are obtained by specifying various acquisition parameters such as the platform altitude, the incidence angle, the frequency, the azimuth and slant range resolutions, and some forest stand properties, including the stand area and density, the tree species and their mean height.

For this study, pine tree forests of 5, 6, 12, 15, 21, 25 and 32 years old are simulated. This age range has been chosen in order to mimic the real dataset available for this test (see Section 3.4.1.2). The platform altitude is set to 3580 meters, corresponding to an airborne system, while the frequency is fixed at 1.3 GHz (L-band). In order to find the best airborne configuration, two experiments are considered. In the first case, the incidence angle is chosen to be $45°$ and the influence of the spatial resolution on classification performance is evaluated. Five datasets are simulated at a resolution of 0.5, 1, 2, 3 and 5 meters. In the second case, the image resolution is fixed to 0.5 meters and several incidence angles are tested: $25°$, $35°$, $45°$ and $55°$.

In both cases, the stand density (D) and the mean tree height ($\bar{H}$) are set according to the desired stand age, as mentioned in Table 3.1.

Stand age a (years)	5	6	12	15	21	25	32
Mean tree height $\bar{H}$ (m)	5.5	6.5	11.6	13.7	17.3	19.2	21.9
Stand density D (stems/ha)	1200	1200	800	800	400	400	300

Table 3.1: Maritime pine stand density D (stems/ha) and mean tree height $\bar{H}$ (m) as a function of stand age (years) (Source: [Ilea *et al.* 2015c] © [2015] IEEE).

The values of the stand density are chosen to be equal to those given by the *Centre Régional de la Propriété Forestière Aquitaine*, for maritime pine forests [CRP 2008], while the mean tree height $\bar{H}$ is obtained by using the Maugé theoretical model given by [Maugé 1987]:

$$\bar{H} = H_{max}(1 - 0.96^a), \tag{3.16}$$

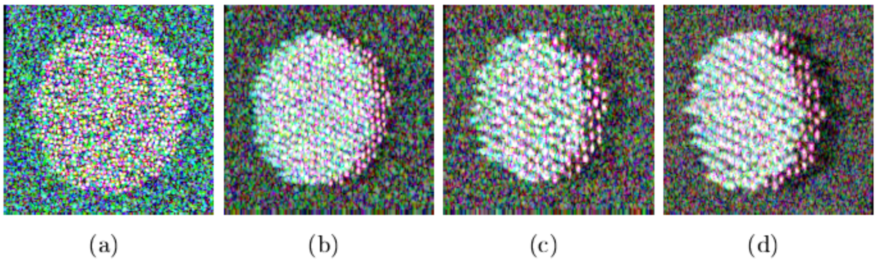
where $H_{max} = 30$ meters is the maximum height and a is the stand age.

By using these numerical values, a database of 350 images is created for each experiment and structured in 4 classes, according to the stand age:

- **1$^{\text{st}}$ class:** less than 10 years (Figure 3.5.a);
- **2$^{\text{nd}}$ class:** between 10 and 20 years (Figure 3.5.b);
- **3$^{\text{rd}}$ class:** between 20 and 30 years (Figure 3.5.c);
- **4$^{\text{th}}$ class:** over 30 years (Figure 3.5.d).

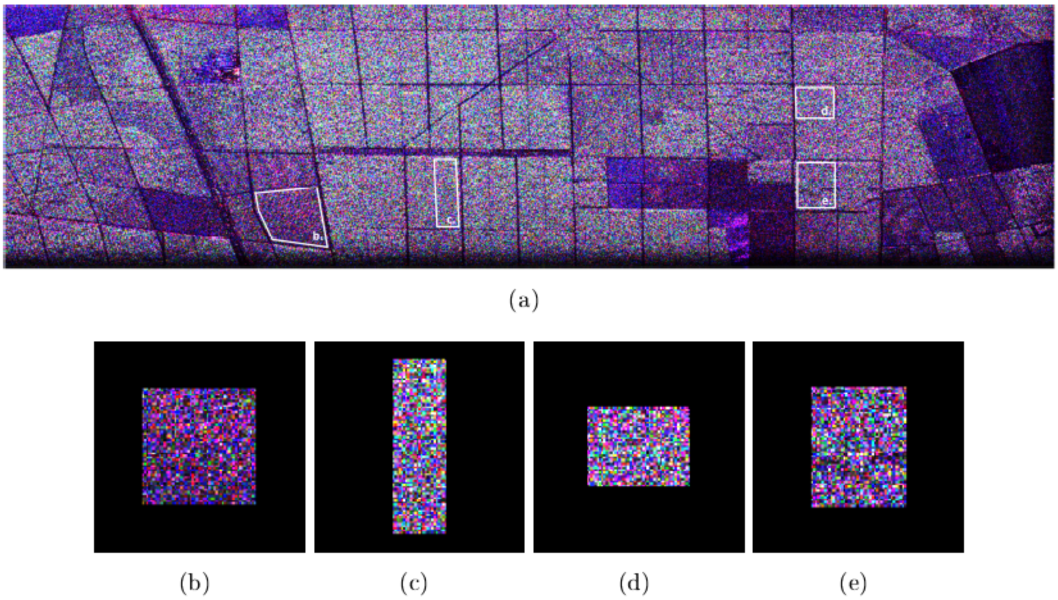
3.4.1.2 Real L-band PolSAR Image

The real L-band PolSAR data displayed in Figure 3.6 consists in one fully polarimetric image (1 meter resolution) acquired on the Nezer maritime pine forest in France, during an ONERA RAMSES campaign in 2004. From this image, 62 forest stands between 5 and 48 years old are identified and grouped in 4 classes, as it has been done for the simulated images.



(a) (b) (c) (d)

Figure 3.5: Examples of L-band pine forest images of: (a) 5, (b) 15, (c) 21 and (d) 32 years old simulated with PolSARproSim software for an incidence angle of 45° and a resolution of 1 meter (Source: [Ilea *et al.* 2015c] © [2015] IEEE).



(a)

(b) (c) (d) (e)

Figure 3.6: (a) Real L-band SAR image and examples of pine forest stands of: (b) 5, (c) 15, (d) 21 and (e) 32 years old (Source: [Ilea *et al.* 2015c] © [2015] IEEE).

3.4.2 Methodology

Polarimetric images contain complex values. As a result, each pixel (x, y) in image I is a complex number having a real part $Re(x, y)$ and an imaginary one $Im(x, y)$:

$$I(x, y) = Re(x, y) + i \times Im(x, y), \tag{3.17}$$

where i is the imaginary unit.

For PolSAR image classification, real-valued images can be used. In order to obtained them, the dB transform is applied:

$$I_{dB}(x, y) = 10 \times \log_{10}(|I(x, y)|), \tag{3.18}$$

where $\log_{10}(\,\cdot\,)$ is the logarithm with base 10, and $|I(x, y)| = \sqrt{Re(x, y)^2 + Im(x, y)^2}$ is the number's modulus.

Further on, classification algorithms based on single polarized real-valued images are considered and compared to new proposed approaches, carried out on polarimetric complex-valued images. The proposed classification methods have to capture both the textural and polarimetric information present in PolSAR images. Several strategies for modeling these images and hence, obtaining the corresponding feature vectors are presented next [Ilea *et al.* 2015b, Ilea *et al.* 2015c, Ilea *et al.* 2015a].

3.4.2.1 GLCM

The first method consists in computing the gray level co-occurrence matrix (GLCM), as presented in Chapter 2, for a single polarized real-valued image.

The GLCMs are computed on the image transformed in dB and quantified with 32 gray levels. The number of quantization levels is chosen by taking into consideration the image size. In a Cartesian coordinate system, the GLCMs are functions of two parameters: the distance $\boldsymbol{d}$ between neighboring pixels and the direction $\boldsymbol{\alpha}$. For this study, $\boldsymbol{d}$ varies between $\mathbf{1}$ and $\mathbf{15}$, and $\boldsymbol{\alpha} = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. The Haralick textural descriptors *homogeneity*, *entropy*, and *correlation* [Haralick *et al.* 1973] along with the *mean* of the gray levels in the initial image are extracted and averaged in the four directions to reduce the sensitivity to the stand's orientation [Regniers *et al.* 2015a]. This workflow is illustrated in Figure 3.7.

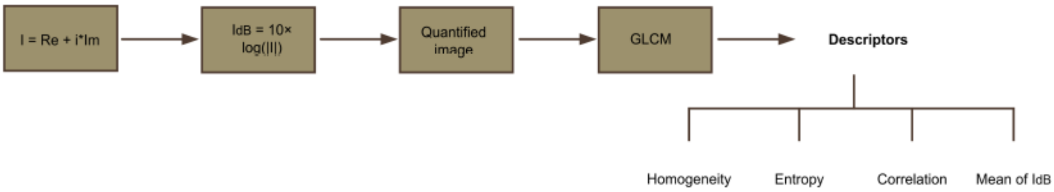


Figure 3.7: GLCM model for a single polarization image.

Later on, in the experimental part, only the channel giving the best results is considered. According to our tests, the amplitude of the HV channel is chosen and the method is denoted by *GLCM HV*.

Next, the proposed classification methods are detailed.

3.4.2.2 MGD Model for a Single Polarization Image

The first approach uses single polarized real-valued images just like the GLCM based algorithm. In this case, the image is transformed in dB and decomposed by using a Daubechies 4 (db4) wavelet transform (WT), with $\mathbf{2}$ scales and $\mathbf{3}$ orientations in order to capture the textural information.

Let S and O be respectively the number of scales and orientations of the wavelet decomposition. Since the subbands of the wavelet decomposition are assumed independent, the square geodesic distance between two images I_1 and I_2 can be expressed as a function of the square geodesic distance computed on each subband as:

$$d^2(I_1, I_2) = \sum_{s=1}^{S}\sum_{o=1}^{O} d^2(\hat{\mathbf{M}}_{1,s,o}, \hat{\mathbf{M}}_{2,s,o}) \tag{3.19}$$

where $\hat{\mathbf{M}}_{1,s,o}$ corresponds to the maximum likelihood estimate of $\mathbf{M}_1$ for the subband at scale s and orientation o.

Next, a 3×3 neighborhood is extracted for each pixel in each subband, capturing the spatial information. Once obtained, the neighborhood's elements are stacked to form a vector with 9 elements. The set of all vectors is then modeled by zero-mean MGDs. The parameter of this distribution, that is the covariance matrix $\hat{\mathbf{M}}$, is estimated by the SCM, or the FP algorithms. The entire process is represented in Figure 3.8.

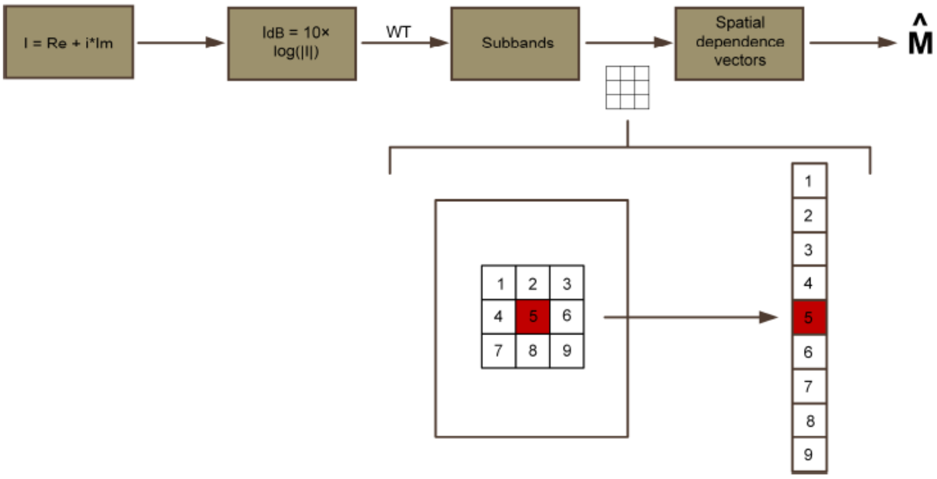


Figure 3.8: MGD model for a single polarization image.

For the experimental part, only the channel giving the best results is considered. In this case, according to our tests, the amplitude of the HH channel is chosen. The method is denoted by *MGD HH + WT + S*, where *WT* states for the wavelet transform and *S* denotes the spatial dependence.

When multiple polarimetric channels are available, the polarimetric dependency can be also exploited, as follows [Ilea *et al.* 2015c].

3.4.2.3 MGD Model for a Three Polarization Image

The HH, HV and VV polarization images are merged into a 3-dimensional array, with each pixel being a complex number. By using this structure, the polarimetric information lying in PolSAR images can be used. Three different algorithms are developed based on:

- *the polarimetric dependence* (denoted *MGD Polar*): the complex 3-dimensional array is modeled by the MGD. For each pixel, the complex information contained in the three polarimetric channels is extracted and organized in a vector with $\mathbf{3}$ elements. The set of all vectors characterizes the cross-channel dependence and it is modeled by the MGD. As a results, a $\mathbf{3} \times \mathbf{3}$ covariance matrix is estimated by using the SCM, or the FP algorithms. This workflow is synthesized in Figure 3.9.

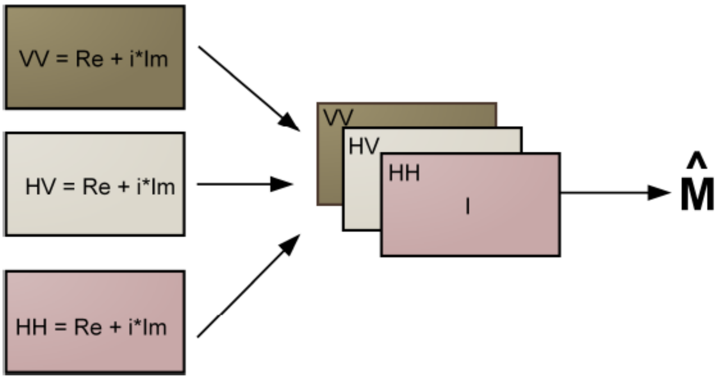


Figure 3.9: Polarimetric dependence.

- *the polarimetric dependence and the wavelet decomposition* (denoted *MGD Polar + WT*): the complex 3-dimensional array is filtered using the db4 wavelet transform with $\mathbf{2}$ scales and $\mathbf{3}$ orientations. For each pixel in each subband, the information contained into the three polarimetric channels is extracted and modeled by the MGD. Thus, a $\mathbf{3} \times \mathbf{3}$ covariance matrix is estimated by using the SCM, or the FP algorithms. The entire algorithm is illustrated in Figure 3.10.

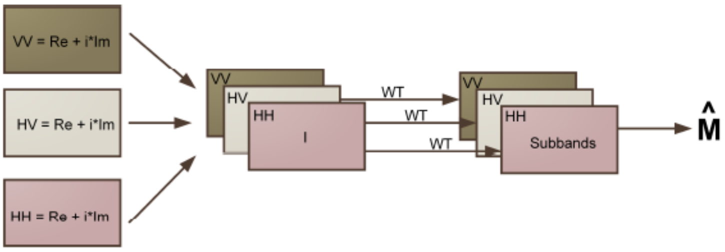


Figure 3.10: Polarimetric dependence and the wavelet decomposition.

- *the polarimetric and spatial dependence, along with the wavelet decomposition* (denoted *MGD Polar + WT + S*): the complex 3-dimensional array is decomposed using the db4 wavelet transform having $\mathbf{2}$ scales and $\mathbf{3}$ orientations. For each pixel in each subband, a spatial dependence given by a $\mathbf{3} \times \mathbf{3}$ neighborhood is considered. The $\mathbf{27}$ extracted elements are structured in a vector and the set of all vectors is modeled by the MGD, as shown in Figure 3.11. A $\mathbf{27} \times \mathbf{27}$ covariance matrix is then estimated with the SCM, or the FP estimator.

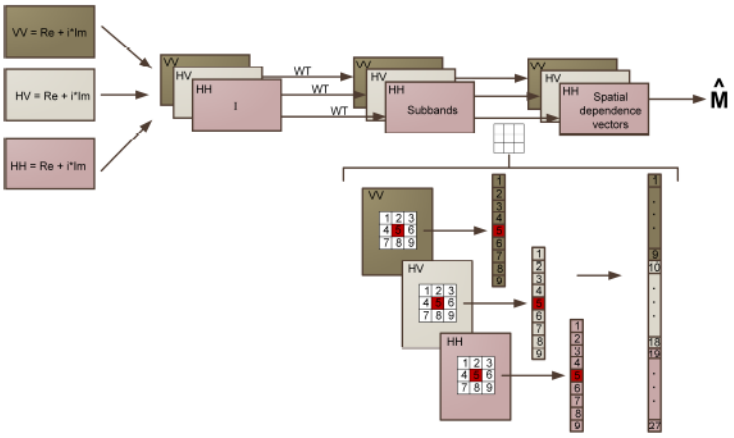


Figure 3.11: Polarimetric and spatial dependence, along with wavelet decomposition.

3.4.3 Results

In the context of a supervised classification approach, the database is randomly divided into a training and a testing set by a cross-validation procedure. The partitioning algorithm is repeated 100 times and, for each iteration, half of the database is used for training, while the other half is used for testing. Once the feature vectors are extracted for all the images, a similarity measure between testing and training images is computed by using the Mahalanobis distance for the GLCM algorithm and the statistic S_{GD}, defined in Section 3.3.1, for the others. All the previously described algorithms are tested and the retrieval performance is evaluated by means of the overall accuracy computed for a k-nearest neighbor classifier (k-NN), with k set to 5.

In the following, the classification performances obtained on both simulated and real SAR images are presented.

3.4.3.1 Simulated L-band SAR Images

As mentioned in Section 3.4.1.1, two types of experiments are performed on simulated data, in order to study the impact of the acquisition parameters on the classification. The considered experiments are designed to find out the best airborne configuration (resolution, incidence angle, number of polarimetric channels) for maritime pine classification according to the stand age. In addition, the relation between the number of polarimetric channels, resolution and classification performance is also studied. In other words, the trade-off between having a single high resolution SAR image, or a low resolution PolSAR image with two, or three channels is addressed.

Influence of the image resolution

First, the influence of the image resolution is tested. For this experiment, the incidence angle is fixed to 45° and the image resolution varies from 0.5m to 5m.

Figure 3.12 draws the influence of distance d to find its best value for the GLCM method. It can be seen that distances between 1 and 5 pixels give the best results.

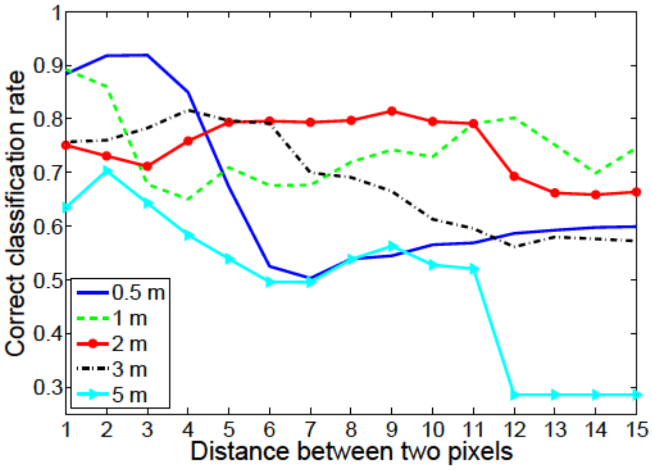


Figure 3.12: Influence of distance d in GLCM on classification accuracy for different spatial resolution (HV channel) (Source: [Ilea *et al.* 2015c] © [2015] IEEE).

Next, Figure 3.13 shows a comparison between the GLCM algorithm and the statistical based approaches, knowing that each time, the polarization with the best performance is retained. In addition, both SCM (Figure 3.13.a) and FP (Figure 3.13.b) estimators are used. By analyzing these results it can be noticed that for simulated data it is better to have one very high resolution polarization channel ($99 \pm 1\%$ for MGD HH + WT + S at 0.5 meters) than a low resolution fully polarimetric SAR image ($85 \pm 4.5\%$ for MGD Polar at 5 meters). For this example, a significant gain of about 14 points is observed. Further on, for high resolution images, the FP estimate improves the classification results over the SCM estimate.

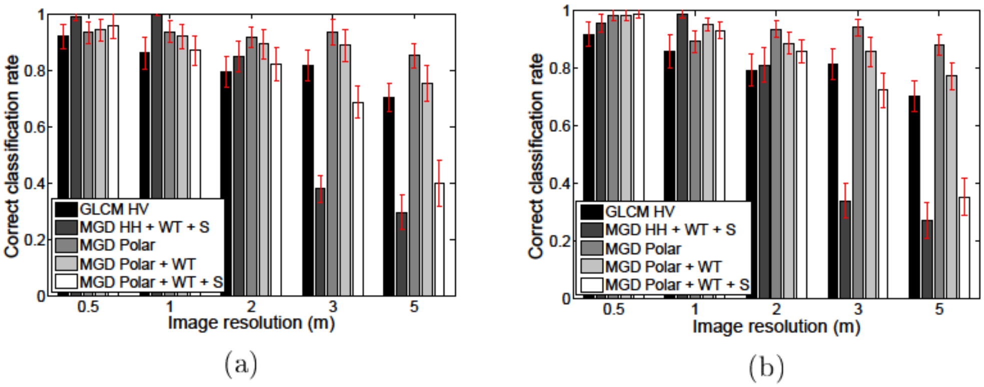


Figure 3.13: Influence of the spatial resolution on classification accuracy for simulated L-band SAR images with incidence angles of 45°, knowing that (a) the SCM and (b) the FP methods are used for the covariance matrix estimation.

Influence of the incidence angle

Second, the influence of the incidence angle is analyzed. For this experiment, the image resolution is fixed to 0.5m and several incidence angles are considered. Like in the previous case, tests are performed to find the appropriate distance d for the GLCM algorithm and the results are shown in Figure 3.14. The best classification rates are retained and compared in Figure 3.15 with those given by the statistical based methods. In addition, both the SCM (Figure 3.15.a) and the FP (Figure 3.15.b) estimators are used. As it can be seen, the GLCM HV is influenced by the incidence angle, while some small changes can be spotted for the other methods.

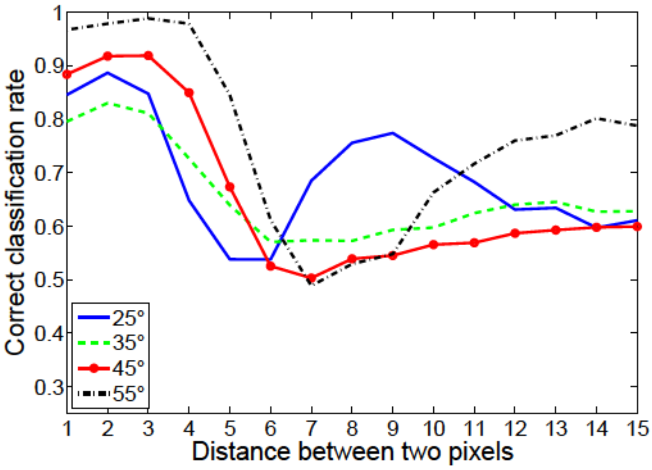


Figure 3.14: Influence of distance d in GLCM on classification accuracy for different incidence angles (HV channel).

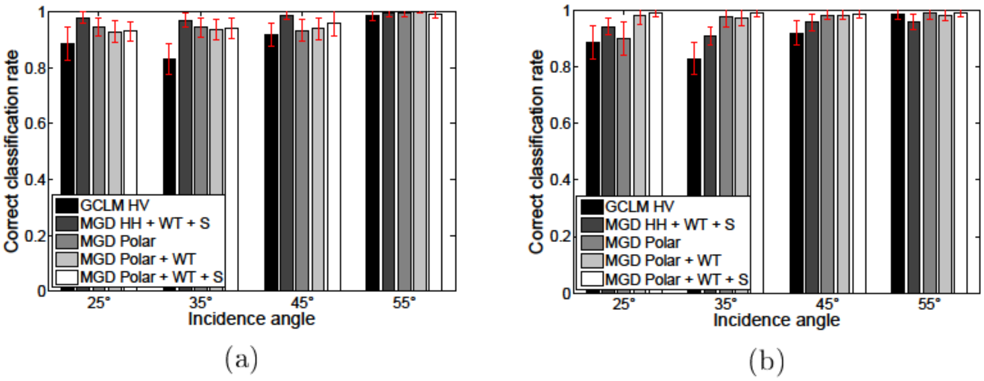


Figure 3.15: Influence of the incidence angle on classification accuracy for simulated L-band SAR images having a resolution of 0.5 meter, knowing that (a) the SCM and (b) the FP methods are used for the covariance matrix estimation.

3.4.3.2 Real L-band SAR Images

Even though PolSARproSim provides a fair level of realism, significant differences can be observed between simulated (Figure 3.5) and real data (Figure 3.6). Those differences are the results of various phenomena, such as forest management practices (thinning operations, plantation density) and natural hazards (storm damages), yielding to some within-class diversity. Hence, as displayed in Table 3.2, classification results on real SAR images are lower than those shown in Section 3.4.3.1 on synthetic dataset. Similar to the case of simulated images with a resolution of 3 and 5m, the best results are given for GLCM HV ($86.6 \pm 5.6\%$) and MGD Polar ($84.0 \pm 4.4\%$) methods. Based on the fact that for these resolutions, the SCM and FP estimator perform very similar, only the results obtained by using the SCM have been reported.

Classification method	Overall accuracy
GLCM HV	$\mathbf{86.6 \pm 5.6}$
MGD HH + WT + S	59.0 ± 5.4
MGD Polar	$\mathbf{84.0 \pm 4.4}$
MGD Polar + WT	81.8 ± 4.0
MGD Polar + WT + S	63.5 ± 4.9

Table 3.2: Comparison between the classification algorithms for real L-band SAR images, knowing that the SCM method is used for the covariance matrix estimation.

3.4.3.3 Results Synthesis

To synthesize the results reported in this section, the following conclusions can be expressed concerning the acquisition parameters. For high resolution images, the link between the forest structure variables (stand age and density, tree height, diameter of tree crown, etc) and the image texture can be exploited. On the other hand, for low resolution images, the textural information is no more visible, but the polarimetric information can be useful in classification. These observations have been confirmed by the experiments performed on simulated data. In this case, the results have shown that it is better to have one very high resolution polarization channel than a low resolution fully polarimetric SAR image. Due to the presence of within-class diversity, this observation can be slightly modified for real PolSAR data.

3.5 Influence of a PDE Based Filtering on PolSAR Image Classification

3.5.1 SAR Images and Speckle Noise

SAR images are characterized by a granular noise pattern, called speckle. This noise is due to the roughness of the analyzed surface with respect to the radar wavelength.

More precisely, in order to obtain SAR images, the target region is illuminated with microwave pulses and the returned signal is recorded. This echo consists of the reflected waves corresponding to the scatterer elements contained in a resolution cell. Since the location of scatterers varies, the received waves are coherent in frequency, but not in phase, causing a pixel-to-pixel variation in intensity, known as the speckle [Lee & Pottier 2009].

From the image processing point of view, this type of noise makes it difficult to analyze SAR images, having a negative impact on the accuracy of image segmentation, or classification [Lee & Pottier 2009]. In order to develop efficient methods to deal with the speckle, its statistical properties have to be taken into account. Therefore, in [Lee 1980] it has been shown that the speckle can be described in terms of a multiplicative noise model:

$$I(x, y) = R(x, y) \times N(x, y), \tag{3.20}$$

where $I(\cdot)$ is the acquired SAR image, $R(\cdot)$ is the noise free reflectance, $N(\cdot)$ is the noise and (x, y) is the considered pixel. In the same work, he proposed the use of the local mean and local variance, for image filtering. This method has been extended next, in [Lee 1981], by using the local gradient information. Being defined on a sliding window, these methods may be influenced by the window's size and form. In [Vasile *et al.* 2006], another approach, has been introduced. By using the region growing algorithm, the authors proposed the construction of adaptive neighborhoods for the averaging process. Different filtering methods, based on non-local means [Zhong *et al.* 2014,Deledalle *et al.* 2015], or on partial differential equation [Yu & Acton 2002] have been also used for image filtering.

The advantage of partial differential equation (PDE) based algorithms lies in the possibility of defining adaptive diffusion functions. These functions allow both smoothing and edge preservation, or even edge enhancement. Therefore, in the following, a PDE based algorithm, more precisely the one introduced in [Terebes *et al.* 2015, Terebes *et al.* 2016], will be considered for PolSAR image denoising.

3.5.2 Noise Removal Algorithm Using Directional Diffusion

In [Terebes *et al.* 2015, Terebes *et al.* 2016], we propose a new directional diffusion method for speckle filtering, based on the multiplicative gradient for edge detection [Mora *et al.* 2012]. This filtering technique is a PDE based approach, that iteratively regularizes the PolSAR images, by updating their values at each position on a discrete two-dimensional grid.

3.5.2.1 Model

In order to develop the mathematical model of the proposed filtering algorithm, the input vector $\mathbf{C}$ is first introduced:

$$\mathbf{C} = \left[|HH|^2 \quad 2|HV|^2 \quad |VV|^2 \right], \tag{3.21}$$

where $|\cdot|$ represents the modulus, while HH, HV and VV are the three complex-valued channels available for PolSAR data. It has to be mentioned that this method can be applied on all the elements in the polarimetric covariance matrix.

Next, starting from $\mathbf{C}$, the span of the PolSAR data is computed, serving as the support for the construction of the multiplicative gradient. For a given iteration t, the span is expressed, at the spatial position (x, y) as:

$$\mathbf{S}(x, y, t) = \sum_{i=1}^{3} \mathbf{C}_i(x, y, t), \tag{3.22}$$

with $\mathbf{C}_i$ being the elements of vector $\mathbf{C}$. For simplification, $\mathbf{S}(x, y, t)$ is denoted further by $\mathbf{S}$. With this notation and based on the procedure presented in [Terebes *et al.* 2015], the norm of the multiplicative gradient $\|\nabla_\eta \mathbf{S}\|$ can be approximated. The approximation is performed considering that the span $\mathbf{S}$ has been spatially discretized on an equally sampled grid, resulting that [Mora *et al.* 2012]:

$$\|\nabla_\eta \mathbf{S}\| = \exp\left(\sqrt{\ln^2\left(\frac{S_{m+1,n}}{S_{m,n}}\right) + \ln^2\left(\frac{S_{m,n+1}}{S_{m,n}}\right)}\right), \tag{3.23}$$

where m and n are points on the discrete grid and $\ln(\cdot)$ is the natural logarithm.

In practice, a smoothed version of the expression in (3.23) is used, by convolving $\mathbf{S}$ with the Gaussian kernel G_σ of standard deviation σ. The use of $\|\nabla_\eta(G_\sigma * \mathbf{S})\|$ is motivated by the need of a noise robust operator.

Further on, the PDE based filter is defined by smoothing along two axes u and v that captures the geometry of all the polarimetric channels. These two axes are obtained starting from the structure tensor proposed in [Di Zenzo 1986] by using the elements C_i of $\mathbf{C}$:

$$G_\rho * \sum_{i=1}^{3} \nabla C_i \left(\nabla C_i\right)^T, \tag{3.24}$$

where G_ρ is a Gaussian kernel function of standard deviation ρ. More precisely, u and v are the eigenvectors of the structure tensor, corresponding to its smallest and largest eigenvalues.

In the end, the PDE based filter can be expressed as:

$$\frac{\partial C_i}{\partial t} = \frac{\partial}{\partial v}\left[g^v\left(\|\nabla_\eta(G_\sigma * S)\|\right) C_{i_v}\right] + \frac{\partial}{\partial u}\left[g^u\left(\|\nabla_\eta(G_\sigma * S)\|\right) C_{i_u}\right], \tag{3.25}$$

where g^v and g^u are the diffusion, or smoothing, functions. In order to have an adaptive smoothing along u and still preserving the information along v, these functions are chosen as proposed in [Tsiotsios & Petrou 2013]:

$$g^v(s) = \exp\left\{-\left(\frac{s}{K_v}\right)^2\right\} \tag{3.26}$$

and

$$g^u(s) = \frac{1}{1 + \frac{s^2}{K_u^2}}, \tag{3.27}$$

with K_v and K_u being the diffusion thresholds along the v and u axes. In addition, for each element C_i the directional derivatives along v and u are defined:

$$C_{i_v} = \frac{\partial C_i}{\partial v} \tag{3.28}$$

and

$$C_{i_u} = \frac{\partial C_i}{\partial u}. \tag{3.29}$$

Due to the use of two different smoothing functions, small-scale coherent structures can appear in the evolving image. These artifacts can be reduced by incorporating the orientation noise in the proposed PDE. Therefore, for each pixel, the orientation given by the structure tensor in (3.24) is modeled by a random process having the following π-periodic probability density function:

$$p(\theta|\theta_m, \sigma_\theta) \propto \frac{1}{\sqrt{2\pi}\sigma_\theta} \exp\left\{ -\frac{(\theta-\theta_m)^2}{2\sigma_\theta^2} \right\}, \tag{3.30}$$

where $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. θ_m is the variable's mean given by the eigenvector corresponding to the smallest eigenvalue of the structured tensor in (3.24) and σ_θ is the variable's standard deviation defined as:

$$\sigma_\theta = \alpha \left(1 - \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right), \tag{3.31}$$

where λ_1 and λ_2 are the largest and smallest eigenvalues of the structure tensor. α represents the maximum variance of the distribution modeling the orientation estimation process and it is measured in degrees.

3.5.2.2 Numerical Approximation

The continuous model given in (3.25) can be numerically approximated by using a spatial and a temporal discretization. For the spatial discretization, the image is assumed to be represented on a discrete grid, equally sampled on the directions of u and v. In addition, for the temporal discretization, uniformly distributed discrete moments are considered. As a result, the continuous function $\mathbf{C}(x, y, t)$ is transformed into its discretized version $\mathbf{C}(mh, nh, adt)$, where m and n are the discrete spatial coordinates, h is the distance between two neighboring points, dt is the time discretization step and a is the number of iteration needed to obtain the scale t. For the experimental part, h is considered to be 1 and dt is set to 0.2.

Based on these observations, the following differences are introduced:

$$D_u^{\pm}(C_i) = \pm(C_{i_{m\pm1,n}} - C_{i_{m,n}}) \tag{3.32}$$

and

$$D_v^{\pm}(C_i) = \pm(C_{i_{m,n\pm1}} - C_{i_{m,n}}), \tag{3.33}$$

where the values corresponding to $(m \pm 1, n)$ and $(m, n \pm 1)$ are obtained by means of biquadratic interpolations [Terebes *et al.* 2004].

In the end, the expression in (3.25) can be approximated by:

$$\begin{aligned}\frac{\partial C_i}{\partial t} = & \, g^u \left(\|\nabla_\eta^E \left(G_\sigma * \mathbf{S}\right)\| \right) D_u^+ \left(C_i\right) - g^u \left(\|\nabla_\eta^W \left(G_\sigma * \mathbf{S}\right)\| \right) D_u^- \left(C_i\right) \\ & + g^v \left(\|\nabla_\eta^S \left(G_\sigma * \mathbf{S}\right)\| \right) D_v^+ \left(C_i\right) - g^v \left(\|\nabla_\eta^N \left(G_\sigma * \mathbf{S}\right)\| \right) D_v^- \left(C_i\right), \qquad (3.34)\end{aligned}$$

where for $h = 1$:

$$\|\nabla_\eta^E \left(G_\sigma * \mathbf{S}\right)\| = \frac{\|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m+1,n} + \|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m,n}}{2}, \qquad (3.35)$$

$$\|\nabla_\eta^W \left(G_\sigma * \mathbf{S}\right)\| = \frac{\|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m-1,n} + \|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m,n}}{2}, \qquad (3.36)$$

$$\|\nabla_\eta^N \left(G_\sigma * \mathbf{S}\right)\| = \frac{\|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m,n-1} + \|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m,n}}{2}, \qquad (3.37)$$

$$\|\nabla_\eta^S \left(G_\sigma * \mathbf{S}\right)\| = \frac{\|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m,n+1} + \|\nabla_\eta \left(G_\sigma * \mathbf{S}\right)\|_{m,n}}{2}. \qquad (3.38)$$

3.5.2.3 Parameters

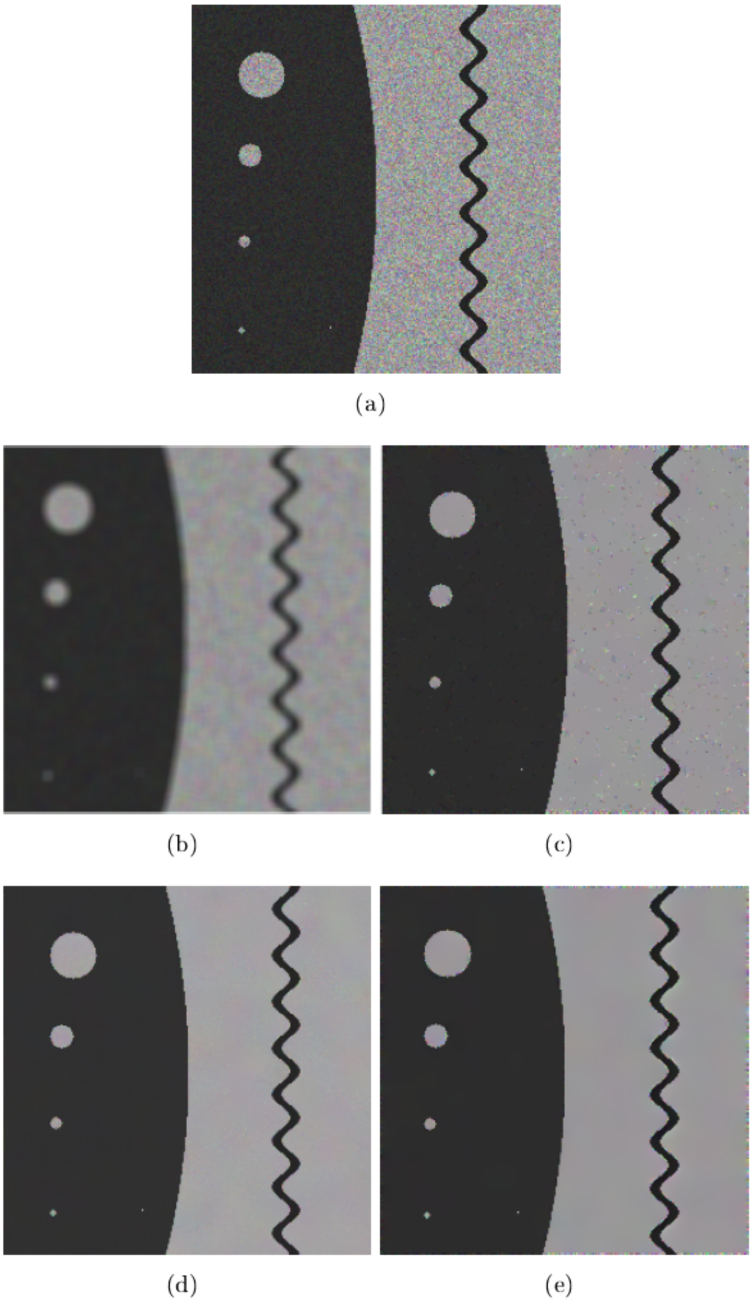
This PDE based filtering method has several parameters that are detailed next:

- the standard deviation σ: characterizes the Gaussian kernel G_σ used for regularization. In practice, the best results have been obtained for values between 0.5 and 1;
- the standard deviation ρ: characterizes the Gaussian kernel G_ρ and it represents the size of the structure tensor. Values between 1 and 3 have been considered in practice;
- the diffusion threshold K_v: characterizes the diffusion along v at each iteration, it depends on the time and the spatial position. In practice it has been computed as a predefined percentage β of the integral value associated to the histogram of (3.23). Typically, β takes values between 0.5 and 1;
- the diffusion threshold K_u: characterizes the diffusion along u. Its value is related to K_v, by considering that $K_u = \gamma K_v$, with $\gamma \geq 1$;
- the standard deviation σ_θ: is a parameter of the probability density function in (3.30) and it is related to parameter α, as expressed in (3.31). This parameter represents the maximum variance of the orientation estimation process and its values are expressed in degrees. In practice, high values give efficiently restored homogeneous regions, but on the other hand, they can degrade the filter's performance on edges. The best results have been obtained for values between 10 and 50.

3.5.2.4 Evaluation

The noise removal algorithm has been tested on both simulated and real PolSAR data, in order to evaluate its performances.

For the first experiment, a synthetic image with Wishart noise has been considered. This image has been created by using the NL-SAR toolbox [Del] and it is shown in Figure 3.16.a. along with the results of several filtering algorithms. Fig-


(a)

(b)

(c)

(d)

(e)

Figure 3.16: Filtering results for a synthetic image: (a) original noisy image, and the filtered images by using (b) the boxcar filter, (c) the SRAD filter [Yu & Acton 2002], (d) the NLSAR filter [Deledalle *et al.* 2015], (e) the proposed method (Source: [Terebes *et al.* 2016] © [2015] IEEE).

ure 3.16.b presents the image obtained by using the boxcar filter and it can be seen that the high frequency content has been eliminated. Figure 3.16.c represents the result given by the SRAD filter [Yu & Acton 2002], which preserves the edges, but it is less efficient for homogeneous regions. Figure 3.16.d shows the results of the NL-SAR filter [Deledalle *et al.* 2015], while Figure 3.16.e contains the results of the proposed method. This last image has been obtained for $\rho = 2.5$, $\sigma = 0.5$. The observation scale has been set to 150 iterations and $dt = 0.2$, $\beta = 0.5$, $\gamma = 2$ and $\sigma_\theta = 40°$

For the second experiment, the Niigata Pi-SAR dataset provided by the PolSARpro software [Pol] has been considered. This image is shown in Figure 3.17.a. Several filters have been used to remove the noise and the results are reported for comparison: the refined Lee filter [Lee 1981] in Figure 3.17.b, the IDAN approach [Vasile *et al.* 2006] in Figure 3.17.c, the non local means based filter [Zhong *et al.* 2014] in Figure 3.17.d and the proposed method in Figure 3.17.e. The last image has been obtained for $\rho = 2.5$, $\sigma = 0.5$. In addition, the observation scale has been set to 15 iterations and $dt = 0.2$, $\beta = 0.7$, $\gamma = 3$.

For the third experiment, the real PolSAR image, presented in Figure 3.6 has been considered. A zoom on this image is shown in Figure 3.18.a and several filtering algorithms have been compared: the Gaussian filter (Figure 3.18.b), the boxcar filter (Figure 3.18.c), an extension of the SRAD filter for PolSAR images (Figure 3.18.d) and the proposed method (Figure 3.18.e).

The parameters of the proposed method are the following. First, the size of the structure tensor ρ is set to 2.5. In addition, the standard deviation σ of the Gaussian kernel used for regularization is computed based on a linear decreasing function, as mentioned in [Whitaker 1993]. The observation scale is set to 5 iterations, $dt = 0.2$, $\beta = 0.25$ and $\gamma = 1.25$.

By analyzing the simulated and real images produced by the proposed method it can be noticed that this filtering algorithm is capable to preserve the high frequency information on edges and textures.

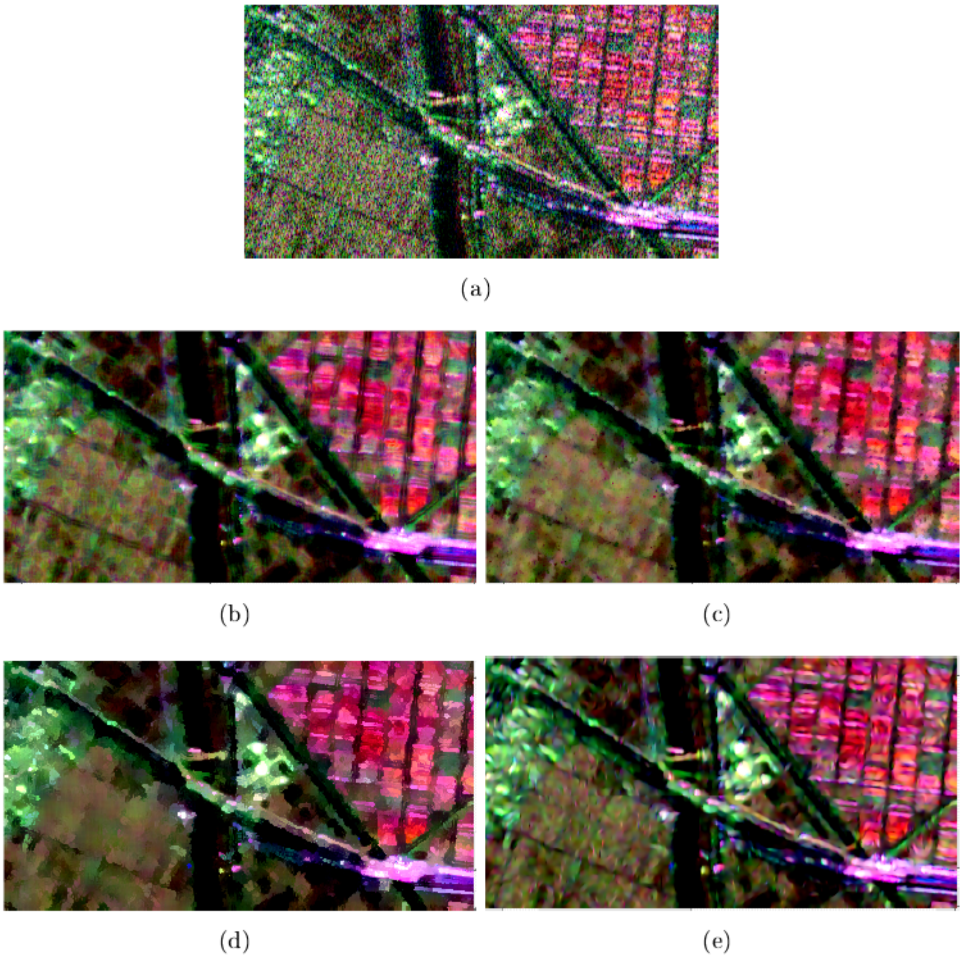
In the next section, the PDE based filtering method is evaluated in the context of PolSAR image classification.

3.5.3 Classification Results

In the following, some of the experiments described in Section 3.4 are repeated on filtered images. The purpose of the performed tests is to study the influence of filtering on the classification.

The classification workflow consists in several steps. First, the amplitude image of each polarization is filtered using the PDE based approach. Next, the covariance matrices are estimated using the SCM algorithm and the MGD model for a single polarization image, introduced in Section 3.4.2.2, is considered.

The tests are carried out on the real L-band SAR image database and the results are reported in Table 3.3. The classification performances are compared to those obtained for no filtered data, but also with those obtained for other filters, namely


(a)

(b)

(c)

(d)

(e)

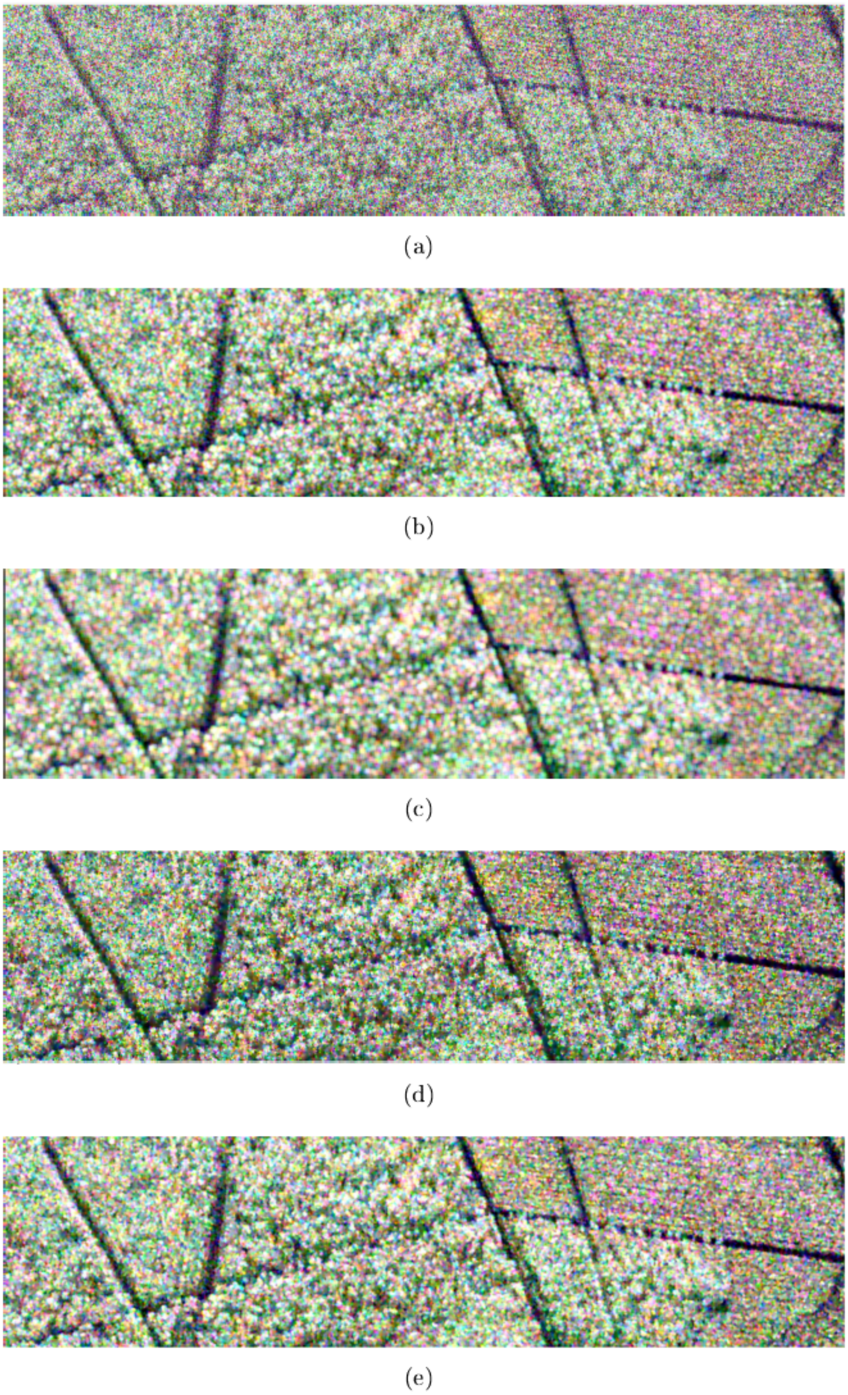
Figure 3.17: Filtering results for the Niigata Pi-SAR PolSAR dataset: (a) original noisy image and the filtered images obtained by using (b) the refined Lee filter [Lee 1981], (c) the IDAN filter [Vasile *et al.* 2006], (d) the non local means based filter [Zhong *et al.* 2014], (e) the proposed method (Source: [Terebes *et al.* 2016] © [2015] IEEE).

the Gaussian filter, the boxcar filter and the SRAD filter. The parameters for all the filters are the same as the ones used to obtain the images in Figure 3.18.

Classification method	Original database	Filtered database			
		Gaussian	Boxcar	SRAD	PDE
MGD HH + WT + S	57.94 ± 6.15	63.00 ± 4.09	62.28 ± 4.24	63.03 ± 5.14	65.47 ± 2.99
MGD HV + WT + S	61.09 ± 5.32	61.38 ± 3.94	62.88 ± 4.64	60.25 ± 6.05	64.47 ± 3.37
MGD VV + WT + S	59.66 ± 4.68	60.94 ± 5.66	65.50 ± 4.68	61.58 ± 5.20	65.91 ± 4.26

Table 3.3: Comparison between the classification performances obtained on non-filtered and filtered real L-band SAR images.

In addition, it has to be mentioned that the classification results obtained for the original database are slightly different from the results reported in Table 3.2.



(a)

(b)

(c)

(d)

(e)

Figure 3.18: Filtering results for the real PolSAR dataset. Zoom on the (a) original noisy image and on the filtered images obtained by using (b) the Gaussian filter with $\sigma = 1$, (c) the boxcar filter of size 5×5, (d) an extension of the SRAD filter for PolSAR images, with 5 iterations and $dt = 0.2$, (e) the proposed method.

This difference is explained by the fact that the training and testing datasets are not the same for the two experiments.

For these parameter choices, it can be noticed that the classification results are improved by filtering the PolSAR image by using the proposed directional diffusion method. Therefore, the PDE based speckle denoising demonstrates its importance as a preprocessing step in the classification workflow.

3.6 Conclusions and Perspectives

3.6.1 Conclusions

In this chapter, a classification algorithm on the space of covariance matrices has been introduced. Several aspects have been addressed.

First, the choice of an appropriate descriptor that takes into account most of the information contained in the images has been discussed. For this purpose, multiscale approaches have been considered. As a result, the images have been decomposed into a set of wavelet subbands.

Second, a distribution capable to model the previously extracted coefficients has been searched. At this point, the use of the zero-mean multivariate Gaussian distribution has been considered to capture the dependencies existing in images, such as textural, or polarimetric dependencies. This model is characterized by its parameter, which is the covariance matrix. By estimating it, the image signature has been obtained. Knowing that robust classification methods are desired, the covariance matrix estimator had to be a robust one. Therefore, a comparison between the sample covariance matrix and the fixed point estimator, also known as the Tyler's estimator, has been carried out in terms of robustness to outliers.

Third, the idea of having a robust decision making strategy for the classification has been addressed. To solve this problem, a statistical hypothesis test has been proposed, based on the geodesic distance. At the beginning, the test has been used along with the MGD model and the SCM estimator. Next, it has been applied to the robust FP estimator and its classification efficiency and noise robustness have been studied.

Further on, the introduced statistic, called S_{GD}, has been applied to PolSAR image classification, on both simulated and real data, illustrating the potential of the proposed classifier. The experiments performed on simulated data have been designed in order to find the best airborne configuration for maritime pine classification according to the stand age. In this context, the results have shown that it is better to have one very high resolution polarization channel than a low resolution fully polarimetric SAR image. Due to the presence of within-class diversity, those conclusions are slightly modified on real PolSAR data.

In the end, a preprocessing step has been added for PolSAR data. This step consists in filtering the speckle that characterizes this type of images, without destroying the textural content. For this purpose, a partial differential equation based algorithm has been proposed. The mathematical formalism and its numerical approximation have been given. The algorithm's results have been qualitatively evaluated on both synthetic and real data. In addition, it has been tested in the context of image classification, showing an improvement of performances.

3.6.2 Perspectives

Further work will include:

- *The extension of the hypothesis test to the case of robust estimators:* in Section 3.3.1, a hypothesis test has been introduced, based on the geodesic distance. The statistic S_{GD} used in the test's definition, has been studied and it has been shown that for the ML estimator of the covariance matrix, it follows a χ^2_{DF} distribution, under H_0. DF denotes the number of freedom degrees and it is equal to the dimension of the parameter space. The hypothesis test has been applied next to the FP estimator, knowing that the statistic's distribution under H_0 has been empirically computed. In order to obtain the theoretical distribution of the test statistic for robust estimators, the number of freedom degrees has to be readjusted, which represents the subject of on going work.

- *The development of an automatic method to tune the parameters of the PDE based filter:* in Section 3.5, a PDE based filtering method has been introduced. This method, has several parameters that have to be specified. In the reported results, these parameters have been tuned case by case, in order to obtain the best results. Future work will address the development of an automatic method to determine the best filter parameters.

Chapter 4

Riemannian Distributions on the Space of Covariance Matrices

Contents

4.1 Introduction

Many works have been dedicated in the literature for the statistical modelling of covariance matrices. Due to its mathematical tractability, the Wishart distribution is certainly the most largely used model in the literature [Wishart 1928, Goodman 1963]. Nevertheless, this model assumes Gaussian statistics for the observations which may not be realistic in practice. More advanced models have hence been proposed based on the so-called scalar product model. These compound models include the $\mathcal{K}$ [Lee *et al.* 1993], $\mathcal{G}^0$ [Freitas *et al.* 2003] and KummerU [Bombrun & Beaulieu 2008] distributions. They have shown promising results notably for the classification of high resolution polarimetric SAR images. Inspired from clustering approaches on Riemannian manifolds [Barachant *et al.* 2013, Nielsen 2013], there is another way to model covariance matrices. By considering Rao's distance on the manifold of covariance matrices, there is a canonical way to define the mean or barycentre of several covariance matrices in this manifold. Based on this concept, the Riemannian Gaussian distribution (RGD) has been introduced to model the statistical variability of real covariance matrices [Said *et al.* 2015b].

This probability density function is characterized by two parameters, its central element and its dispersion around this central element. For this model, the maximum likelihood estimator (MLE) of the central value corresponds to the Riemannian center of mass. While being efficient to model the mean element, this estimator is easily influenced by the presence of aberrant data [Bishop 2007, Afsari 2011, Formont *et al.* 2013]. In practice, outliers may arise from faulty measurements, or they may be explained by the inherent variability of data. To overcome this problem, a robust estimator of the central element can be considered, such as the Riemannian median [Yang 2010, Barbaresco *et al.* 2013, Fletcher *et al.* 2009]. Therefore, we have introduced in [Hajri *et al.* 2016] the Riemannian Laplace distribution (RLD), a generative model for which the MLE of the central element is the Riemannian median. This distribution depends also on two parameters: the central value and the dispersion.

In this chapter, the RGD and the RLD are defined and they are used, in the context of texture image classification. In order to model the within-class diversity, the mixtures of RGDs or RLDs, can be considered. In this case, a new parameter appears that is the mixture's weight. The entire classification workflow is shown in Figure 4.1 and it consists in several steps.

First, starting from the initial database, the covariance matrices representing the image signature are extracted, during the feature extraction stage. This step has been already detailed in Chapter 3. Knowing that supervised classification algorithms are applied, the dataset is divided into two subsets. One of them is used for training, and the other one for testing. Next, the elements in the training set are modeled on the Riemannian manifold by mixtures of RGDs, or RLDs, and the parameters characterizing each image class (the central value, the dispersion and the mixture's weight) are estimated by using algorithms like k-means, or expectation-maximization (EM). To obtain the estimated values, these two well-

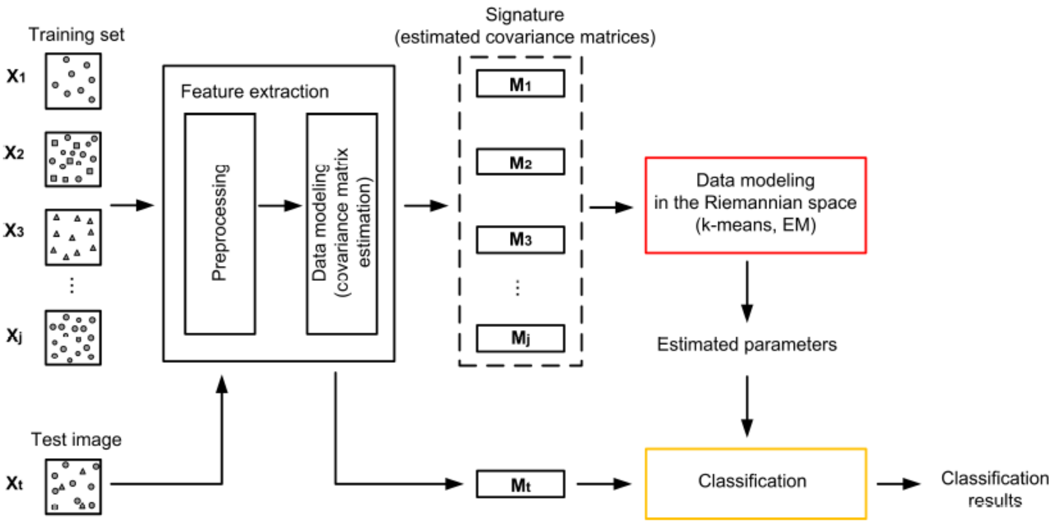

Training set
X1
X2
X3
Xj
Feature extraction
Preprocessing
Data modeling (covariance matrix estimation)
Signature (estimated covariance matrices)
M1
M2
M3
Mj
Data modeling in the Riemannian space (k-means, EM)
Estimated parameters
Test image
Xt
Mt
Classification
Classification results

Figure 4.1: Classification workflow based on Riemannian distribution modeling on the space of covariance matrices.

known clustering methods are adapted for working on the Riemannian manifold. Details on the demanded modifications will be provided later in this chapter. In the end, the classification is performed, by assigning each test observation to the closest training set in terms of a predefined criterion. For this purpose, the classical linear discriminant analysis and quadratic discriminant analysis are generalized to the case of covariance matrices.

The chapter is structured as follows. Section 4.2 introduces some theoretical elements concerning the Riemannian geometry on the manifold of covariance matrices. Section 4.3 and Section 4.4 define respectively the Riemannian Gaussian distribution and the Riemannian Laplace distribution. In addition, the parameter estimation process is detailed. Moreover, the mixture models are described and both the k-means and EM algorithms are detailed for these Riemannian distributions. These algorithms demand the definition of the number of mixture models, or clusters. In order to automatically compute the appropriate value, the Bayesian Information Criterion (BIC) is considered. In Section 4.5, the RGD and RLD are compared for texture image classification and the influence of outliers is analyzed. In the end, Section 4.6 reports some conclusions and perspectives.

4.2 Riemannian Geometry on the Manifold of Covariance Matrices

4.2.1 The Space of Symmetric Positive Definite Matrices

Let $\mathcal{P}_m$ be the space of all $m \times m$ symmetric and positive definite matrices $\mathbf{M} \in \mathbb{R}^{m \times m}$, satisfying the following conditions:

$$\mathbf{M} - \mathbf{M}^T = 0 \tag{4.1}$$

and

$$\mathbf{x}^T\mathbf{M}\mathbf{x} > 0, \tag{4.2}$$

$\forall \mathbf{x} \in \mathbb{R}^m$ and $\mathbf{x} \neq 0$.

In practice, the space $\mathcal{P}_m$ can be represented, for instance, by the space of structure tensors [Rosu *et al.* 2016], diffusion tensors [Fletcher & Joshi 2007,Pennec *et al.* 2006], or even non-degenerate covariance matrices [Said *et al.* 2015b].

4.2.2 Riemannian Geodesic Distance

The ideas of similarity and distance in the space $\mathcal{P}_m$ can be expressed by means of the Rao's distance, or geodesic distance. The geodesic distance between two points $\mathbf{M}_1$ and $\mathbf{M}_2$ on the manifold is given by the length of the shortest curve connecting the two points [Terras 1988,Helgason 2001].

Mathematically, this definition can be stated as follows [Said *et al.* 2015a]. Let $d : \mathcal{P}_m \times \mathcal{P}_m \longrightarrow \mathbb{R}_+$ be the geodesic distance, $\mathbf{M}_1, \mathbf{M}_2 \in \mathcal{P}_m$ and $c : [0, 1] \longrightarrow \mathcal{P}_m$ a differentiable curve, with $c(0) = \mathbf{M}_1$ and $c(1) = \mathbf{M}_2$. Thus, the length of curve c, denoted by $L(c)$ is computed as:

$$L(c) = \int_0^1 \left\| \frac{dc}{dt} \right\| dt \tag{4.3}$$

and the geodesic distance $d(\mathbf{M}_1, \mathbf{M}_2)$ is the infimum of $L(c)$ with respect to all differentiable curves c. Based on the properties of this metric, it has been shown that the unique curve γ fulfilling this condition is:

$$\gamma(t) = \mathbf{M}_1^{\frac{1}{2}} \left(\mathbf{M}_1^{-\frac{1}{2}} \mathbf{M}_2 \mathbf{M}_1^{-\frac{1}{2}} \right)^t \mathbf{M}_1^{\frac{1}{2}}, \tag{4.4}$$

called the geodesic connecting $\mathbf{M}_1$ and $\mathbf{M}_2$. In the end, the geodesic distance becomes [James 1973]:

$$d^2(\mathbf{M}_1, \mathbf{M}_2) = \operatorname{tr}\Big[\log \left(\mathbf{M}_1^{-\frac{1}{2}} \mathbf{M}_2 \mathbf{M}_1^{-\frac{1}{2}} \right) \Big]^2 = \sum_i (\ln \lambda_i)^2, \tag{4.5}$$

with λ_i, $i = 1, \ldots, m$ being the eigenvalues of $\mathbf{M}_2^{-1}\mathbf{M}_1$. This equation is equivalent to the one introduced in (3.12). For simplicity, the constant equaling $\frac{1}{2}$ in (3.12) will be omitted, since this constant can be transferred to the dispersion parameter of the Riemannian distributions.

4.2.3 Riemannian Exponential Mapping and Riemannian Logarithm Mapping

The Riemannian exponential mapping and Riemannian logarithm mapping are two operators that make possible the transition between a point on the manifold $\mathbf{M}_1 \in \mathcal{P}_m$ and the tangent space at that point $T_{\mathbf{M}_1}$. This space contains the vectors V that are tangent to all posible curves passing through $\mathbf{M}_1$, as shown in Figure 4.2.

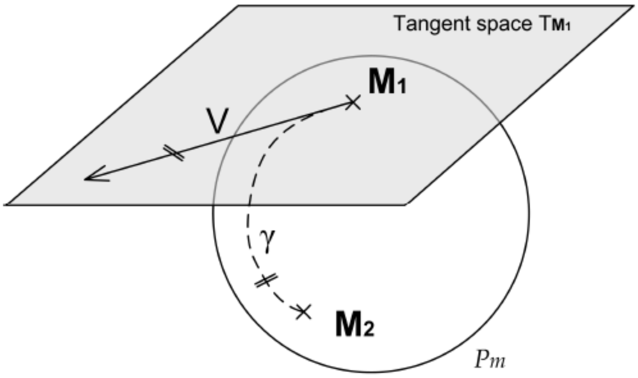


Figure 4.2: Illustration of the tangent space $T_{\mathbf{M}_1}$ at point $\mathbf{M}_1 \in \mathcal{P}_m$.

More precisely, the Riemannian exponential mapping for a point $\mathbf{M}_1 \in \mathcal{P}_m$ and the tangent vector V is given by [Higham 2008, Fletcher *et al.* 2009]:

$$\mathbf{M}_2 = \mathrm{Exp}_{\mathbf{M}_1}(V) = \mathbf{M}_1^{\frac{1}{2}} \exp\left(\mathbf{M}_1^{-\frac{1}{2}} V \mathbf{M}_1^{-\frac{1}{2}}\right) \mathbf{M}_1^{\frac{1}{2}}, \tag{4.6}$$

where $\exp(\cdot)$ is the matrix exponential. By this transformation, the tangent vector V can be mapped on the manifold.

Further on, the inverse of the Riemannian exponential mapping is the Riemannian logarithm mapping. Between two points $\mathbf{M}_1, \mathbf{M}_2 \in \mathcal{P}_m$ this operator is given by [Higham 2008, Fletcher *et al.* 2009]:

$$V = \mathrm{Log}_{\mathbf{M}_1}(\mathbf{M}_2) = \mathbf{M}_1^{\frac{1}{2}} \log\left(\mathbf{M}_1^{-\frac{1}{2}} \mathbf{M}_2 \mathbf{M}_1^{-\frac{1}{2}}\right) \mathbf{M}_1^{\frac{1}{2}}, \tag{4.7}$$

where $\log(\cdot)$ is the matrix logarithm. In practice, this operation gives the tangent vector V, by transforming the geodesic γ in a straight line in the tangent space. In addition, the geodesic's length between $\mathbf{M}_1$ and $\mathbf{M}_2$ is equal to the norm of the tangent vector V.

By using all these theoretical aspects concerning the space $\mathcal{P}_m$, the Riemannian distributions are introduced next.

4.3 Riemannian Gaussian Distributions

4.3.1 Definition

The probability density function of the Riemannian Gaussian distribution (RGD) with respect to the Riemannian volume element, in the space $\mathcal{P}_m$ of $m \times m$ real, symmetric and positive definite matrices, has been introduced in [Said *et al.* 2015b] as:

$$p\left(\mathbf{M}|\bar{\mathbf{M}}, \sigma\right) = \frac{1}{Z(\sigma)} \exp\left\{-\frac{d^2(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2}\right\}, \tag{4.8}$$

where $\bar{\mathbf{M}} \in \mathcal{P}_m$ is the central value and $\sigma \in \mathbb{R}^+$ is the dispersion parameter. $d(\cdot)$ is the Riemannian distance given in (4.5) and $Z(\sigma)$ is a normalization factor independent of $\bar{\mathbf{M}}$. The computation of this factor is detailed next.

4.3.2 Normalization Factor

4.3.2.1 Definition

The normalization factor $Z(\sigma)$ has the following expression [Said *et al.* 2015a]:

$$Z(\sigma) = \int_{\mathcal{P}_m} \exp\left\{-\frac{d^2(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2}\right\} dv(\mathbf{M}), \tag{4.9}$$

where $dv(\mathbf{M})$ is the Riemannian volume element. For the special case, of $m = 2$ this factor has a close form [Said *et al.* 2015a]:

$$Z(\sigma) = (2\pi)^{3/2}\,\sigma^2 \exp(\sigma^2/4)\,\mathrm{erf}(\sigma/2), \tag{4.10}$$

where $\mathrm{erf}(\cdot)$ is the error function, defined as [Lebedev & Silverman 1972]:

$$\mathrm{erf}(t) = \int_0^t \exp(-x^2)dx. \tag{4.11}$$

For larger matrices, that is for $\mathbf{M} \in \mathcal{P}_m$, $m > 2$, the values of $Z(\sigma)$ can be computed by using the Monte Carlo integration technique. Starting from (4.9), it has been shown in [Said *et al.* 2015b] that the normalization factor has the general expression:

$$Z(\sigma) = q_m \times \int_{\mathbb{R}^m} \exp\left\{-\frac{|\mathbf{r}|^2}{2\sigma^2}\right\} \prod_{i<j} \sinh\left\{\frac{|r_i - r_j|}{2}\right\} dr_1 \ldots dr_m, \tag{4.12}$$

where $|\mathbf{r}| = (r_1^2 + \ldots + r_m^2)^{\frac{1}{2}}$ and q_m is given by:

$$q_m = \frac{1}{m!}\,\frac{\pi^{\frac{m^2}{2}}}{\Gamma_m(\frac{m}{2})}\,8^{\frac{m(m-1)}{4}}. \tag{4.13}$$

$\Gamma_m(\cdot)$ is the multivariate Gamma function [Muirhead 1982] defined as:

$$\Gamma_m(y) = \pi^{\frac{m(m-1)}{4}} \prod_{j=1}^{m} \Gamma\left(y + \frac{1-j}{2}\right), \tag{4.14}$$

and $\Gamma(\cdot)$ is the Gamma function.

4.3.2.2 Numerical Computation of $Z(\sigma)$

In practice, the evaluation of the expression in (4.12) is done by sampling $\mathbf{r}$ from a zero-mean multivariate Gaussian distribution of covariance matrix $\sigma^2 I_m$:

$$p\left(\mathbf{x}|\sigma^2 I_m\right) = \frac{1}{(2\pi)^{\frac{m}{2}}\,\sigma^m} \exp\left\{-\frac{\mathbf{x}^T\mathbf{x}}{2\sigma^2}\right\}, \tag{4.15}$$

with I_m being the identity matrix of size $m \times m$. Therefore, the normalization factor in (4.12) becomes:

$$Z(\sigma) = (2\pi)^{\frac{m}{2}} \sigma^m \, E\left[q_m \prod_{i<j} \sinh\left\{ \frac{|x_i - x_j|}{2} \right\} \right], \tag{4.16}$$

where $E[\cdot]$ denotes the expectation with respect to the zero-mean multivariate Gaussian distribution and q_m is given in (4.13). In [Zanini *et al.* 2016] it has been shown that the Monte Carlo integration may lead to instability problems for large values of m. To solve this problem, the authors of have proposed to smooth the results by means of cubic spline functions. In the end, tables containing the values of $Z(\sigma)$ can be built. Some examples are shown in Figure 4.3, where the normalization factor is plotted for three dimensions of covariance matrices $m = 3$, 5 and 10 and σ varying from 0 to 0.7.

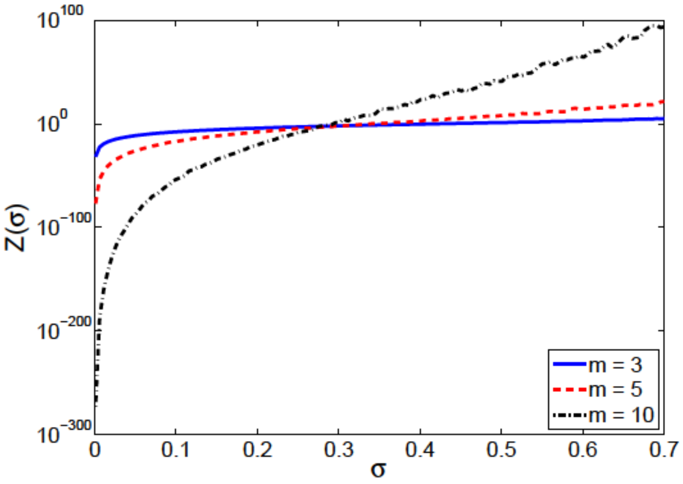


Figure 4.3: Normalization factor $Z(\sigma)$ as a function of dispersion σ for different matrix sizes.

Besides the normalization factor, in the expression of the RGD, there are also two parameters: the central value and the dispersion. The algorithms to estimate them are presented in the next section.

4.3.3 Parameter Estimation

The RGD's parameters, the central value and the dispersion, can be estimated through the maximum likelihood estimation (MLE), as follows.

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$ be a set of N independent and identically distributed (i.i.d.) samples according to a Riemannian Gaussian distribution of central value $\bar{\mathbf{M}}$ and dispersion σ. First, the MLE of the central value $\bar{\mathbf{M}}$ is the Riemannian center of mass $\widehat{\mathbf{M}}$, obtained by minimizing the cost function:

$$f_{CM}(\bar{\mathbf{M}}) = \frac{1}{N}\sum_{n=1}^{N} d^2(\bar{\mathbf{M}}, \mathbf{M}_n), \tag{4.17}$$

where $d(\,\cdot\,)$ is the geodesic distance [James 1973] introduced in (4.5). Details on the computational algorithm can be found further, in Section 5.2.1. Next, the MLE estimate $\hat{\sigma}$ of dispersion σ is given by the solution of:

$$\sigma^3 \times \frac{d}{d\sigma}Z\left(\sigma\right) = f_{CM}(\widehat{\mathbf{M}}), \tag{4.18}$$

with $Z\left(\sigma\right)$ being the normalization factor. In practice, for $m = 2$ the dispersion equation is solved by a conventional Newton-Raphson algorithm [Said *et al.* 2015a]. If $m > 2$, the expression in (4.18) is evaluated by means of Monte Carlo integration. In this case, a table containing the values of $\frac{d}{d\sigma}Z\left(\sigma\right)$ has to be built. The same steps as for the computation of $Z(\sigma)$ are followed. First, the derivative of $Z(\sigma)$ with respect to σ is expressed:

$$\begin{aligned}\frac{d}{d\sigma}Z(\sigma) &= \int_{\mathcal{P}_m} \frac{d^2(\mathbf{M},\bar{\mathbf{M}})}{\sigma^3}\exp\left\{-\frac{d^2(\mathbf{M},\bar{\mathbf{M}})}{2\sigma^2}\right\}dv(\mathbf{M}) \\ &= q_m \times \int_{\mathbb{R}^m} \frac{|\mathbf{r}|^2}{\sigma^3}\exp\left\{-\frac{|\mathbf{r}|^2}{2\sigma^2}\right\}\prod_{i<j}\sinh\left\{\frac{|r_i - r_j|}{2}\right\}dr_1 \dots dr_m,\end{aligned} \tag{4.19}$$

where $|\mathbf{r}| = (r_1^2 + \ldots + r_m^2)^{\frac{1}{2}}$ and q_m is given in (4.13). Next, vector $\mathbf{r}$ is sampled from a multivariate Gaussian distribution and the final expression is achieved:

$$\frac{d}{d\sigma}Z(\sigma) = (2\pi)^{\frac{m}{2}}\sigma^m\ E\left[q_m\,\frac{|\mathbf{x}|^2}{\sigma^3}\prod_{i<j}\sinh\left\{\frac{|x_i - x_j|}{2}\right\}\right]. \tag{4.20}$$

with $E[\,\cdot\,]$ being the expectation with respect to the zero-mean multivariate Gaussian distribution of covariance matrix $\sigma^2 I_m$.

4.3.4 Mixture Model for RGDs

Earlier in this section, the RGDs have been introduced in (4.8). Their definition has been also generalized for mixture models.

Let $\mathcal{M} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be a sample of N i.i.d observations modeled by a mixture of K Riemannian Gaussian distributions. Starting from (4.8), the probability density function for a mixture of K RGDs can be defined as [Said *et al.* 2015b]:

$$p\left(\mathbf{M}|\theta\right) = \sum_{k=1}^{K}\varpi_k\, p\left(\mathbf{M}|\bar{\mathbf{M}}_k, \sigma_k\right), \tag{4.21}$$

where $\theta = \left\{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1\leq k\leq K}\right\}$ is the parameter vector. ϖ_k are positive weights, with $\sum_{k=1}^{K}\varpi_k = 1$ and $p\left(\mathbf{M}|\bar{\mathbf{M}}_k, \sigma_k\right)$ is given by (4.8).

For each component $k = 1, \dots, K$, the parameters $\hat{\theta} = \{(\hat{\varpi}_k, \widehat{\bar{\mathbf{M}}}_k, \hat{\sigma}_k)_{1 \leq k \leq K}\}$ can be estimated by using several algorithms, like k-means, or the expectation maximization. These estimation methods are detailed next.

4.3.4.1 Parameters Estimation by Using the K-means Algorithm

The simplest estimation method implies the computation of centroids $\widehat{\bar{\mathbf{M}}}_k$, of clusters c_k, $k = 1, \dots, K$ by using the intrinsic k-means algorithm on the Riemannian manifold [Faraki *et al.* 2015a]. Thus, for each cluster c_k, the cost function

$$f_{CM}(\bar{\mathbf{M}}_k) = \frac{1}{N_k} \sum_{n=1}^{N_k} d^2\left(\bar{\mathbf{M}}_k, \mathbf{M}_{k_n}\right) \tag{4.22}$$

has to be minimized, where $\mathbf{M}_{k_n} \in c_k$, $n = 1, \dots, N_k$ and N_k is the cardinal of c_k. The minimizer of the cost function defined in (4.22) is known to be the Riemannian centre of mass of c_k. The estimation procedure is repeated for a fixed number of iterations N_{max}, or until its convergence, that is until the values remain almost stable for successive iterations.

Next, once that centroids $\widehat{\bar{\mathbf{M}}}_k$ are determined, for each cluster c_k, the estimated dispersion parameter $\hat{\sigma}_k$ is obtained as the solution of:

$$\sigma_k^3 \times \frac{d}{d\sigma_k} Z\left(\sigma_k\right) = f_{CM}(\widehat{\bar{\mathbf{M}}}_k), \tag{4.23}$$

where $Z\left(\sigma_k\right)$ is the normalization factor for the k^{th} mixture. As mentioned earlier, this latter is solved by a conventional Newton-Raphson algorithm [Said *et al.* 2015a] for $m = 2$ and by using the Monte Carlo integration, if $m > 2$.

Finally, the estimated weights $\hat{\varpi}_k$ are given by:

$$\hat{\varpi}_k = \frac{N_k}{\sum_{k=1}^{K} N_k}. \tag{4.24}$$

All the estimation procedure is synthesized in Algorithm 1.

4.3.4.2 Parameters Estimation by Using the Expectation-Maximization Algorithm

The second approach for the estimation of parameters $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1 \leq k \leq K}\}$ implies the use of the expectation-maximization algorithm (EM), introduced in [Said *et al.* 2015a] for mixtures of RGDs. In their work, the EM algorithm has been extended to the Riemannian case, as follows.

First, two quantities are defined for each mixture component k, $k = 1, \dots, K$:

$$\omega_k(\mathbf{M}_n, \theta) = \frac{\varpi_k \times p(\mathbf{M}_n | \bar{\mathbf{M}}_k, \sigma_k)}{\sum_{s=1}^{K} \varpi_s \times p(\mathbf{M}_n | \bar{\mathbf{M}}_s, \sigma_s)} \tag{4.25}$$

and

$$n_k(\theta) = \sum_{n=1}^{N} \omega_k(\mathbf{M}_n). \tag{4.26}$$

Next, the estimated parameters $\hat{\theta} = \{(\hat{\varpi}_k, \widehat{\mathbf{M}}_k, \hat{\sigma}_k)_{1 \leq k \leq K}\}$ are iteratively updated based on the current value of $\hat{\theta}$:

- The mixture weight $\hat{\varpi}_k$ is given by:

$$\hat{\varpi}_k = \frac{n_k(\hat{\theta})}{\sum_{k=1}^{K} n_k(\hat{\theta})}; \tag{4.27}$$

- The central value $\widehat{\mathbf{M}}_k$ is computed as:

$$\widehat{\mathbf{M}}_k = \arg\min_{\mathbf{M}} \sum_{n=1}^{N} \omega_k(\mathbf{M}_n, \hat{\theta}) d^2(\mathbf{M}, \mathbf{M}_n); \tag{4.28}$$

- The dispersion $\hat{\sigma}_k$ is obtained as:

$$\hat{\sigma_k} = \Phi(n_k^{-1}(\theta)) \times \sum_{n=1}^{N} \omega_k(\mathbf{M}_n, \hat{\theta}) d^2(\widehat{\mathbf{M}}_k, \mathbf{M}_n), \tag{4.29}$$

where Φ is the inverse function of $\sigma \mapsto \sigma^3 \times \frac{d}{d\sigma} \log Z(\sigma)$.

It has to be mentioned that the order of the above steps has to be respected to obtain the estimated parameters, i.e. convergence is ensured.

Similar to the k-means algorithm, the estimation procedure is repeated for a fixed number of iterations N_{max}, or until the values remain almost stable for successive iterations, that is the algorithm's convergence. The estimation procedure is synthesized in Algorithm 2.

4.3.4.3 Bayesian Information Criterion

The k-means and EM algorithms are implemented based on a predefined parameter, that is the number of mixture components K. In order to circumvent this drawback, the Bayesian Information criterion (BIC) can be used. In [Prendes *et al.* 2015], the authors considered another method, that is a Bayesian non parametric approach through a Dirichlet process mixture (DPM) model to estimate the number of components in the mixture model. In the following, the BIC will de considered. This criterion has been introduced in [Schwarz 1978], and it represents a method to automatically find the best value of K for fitting the data.

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be a sample of N i.i.d observations modeled by a mixture of K Riemannian Gaussian distributions, where K is unknown. By using the BIC, the estimated value $\hat{K}$ is obtained according to:

$$\hat{K} = \arg\min_{K} BIC(K), \tag{4.30}$$

Algorithm 1 K-means estimation algorithm for mixtures of K RGDs

```
Input: $\mathbf{M}_1, \ldots, \mathbf{M}_N$, $K$, $N_{\max}$
for $k = 1 : K$ do
    Initialize $\bar{\mathbf{M}}_k$ randomly.
    for $n = 1 : N$ do
        Assign $\mathbf{M}_n$ to its closest centroid $\bar{\mathbf{M}}_k$.
    end for
end for
$it = 1$.
repeat
    for $k = 1 : K$ do
        Estimate $\bar{\mathbf{M}}_k$ according to (4.22).
    end for
    for $n = 1 : N$ do
        Assign $\mathbf{M}_n$ to its closest centroid $\bar{\mathbf{M}}_k$.
    end for
    $it = it + 1$.
until (convergence) or ($it > N_{\max}$)
for $k = 1 : K$ do
    Estimate $\sigma_k$ according to (4.23).
    Estimate $\varpi_k$ according to (4.24).
end for
Output: $\hat{\theta} = \{(\hat{\varpi}_k, \widehat{\mathbf{M}}_k, \hat{\sigma}_k)_{1 \leq k \leq K}\}$
```

Algorithm 2 Expectation-maximization estimation algorithm for mixtures of K RGDs

```
Input: $\mathbf{M}_1, \ldots, \mathbf{M}_N$, $K$, $N_{\max}$
for $k = 1 : K$ do
    Initialize $\varpi_k$ with $\frac{1}{K}$.
    Initialize $\bar{\mathbf{M}}_k$ randomly.
    Initialize $\sigma_k$ with the solution of (4.23).
end for
$it = 1$.
repeat
    for $k = 1 : K$ do
        Estimate $\varpi_k$ according to (4.27).
        Estimate $\bar{\mathbf{M}}_k$ according to (4.28).
        Estimate $\sigma_k$ according to (4.29).
    end for
    $it = it + 1$.
until (convergence) or ($it > N_{\max}$)
Output: $\hat{\theta} = \{(\hat{\varpi}_k, \widehat{\mathbf{M}}_k, \hat{\sigma}_k)_{1 \leq k \leq K}\}$
```

where

$$BIC(K) = -LL + \frac{1}{2} \times DF \times \log(N). \tag{4.31}$$

In the previous expression, LL is the log-likelihood given by:

$$LL = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \hat{\varpi}_k p(\mathbf{M}_n | \widehat{\bar{\mathbf{M}}}_k, \hat{\sigma}_k) \right\}, \tag{4.32}$$

where $(\hat{\varpi}_k, \widehat{\bar{\mathbf{M}}}_k, \hat{\sigma}_k)_{1 \leq k \leq M}$ are obtained by using the k-means, or the EM algorithm described earlier in this section, assuming that the exact dimension is K. Moreover, DF is the number of degrees of freedom of the statistical model defined as:

$$DF = K \times \frac{m(m+1)}{2} + K + (K-1). \tag{4.33}$$

In this expression, $K \times \frac{m(m+1)}{2}$ corresponds to the number of freedom degrees associated to $(\widehat{\bar{\mathbf{M}}}_k)_{1 \leq k \leq K}$, K corresponds to $(\hat{\sigma}_k)_{1 \leq k \leq K}$ and $K-1$ corresponds to $(\hat{\varpi}_k)_{1 \leq k \leq K})$, knowing that $\sum_{k=1}^{K} \hat{\varpi}_k = 1$.

4.4 Riemannian Laplace Distributions

As mentioned earlier, in Section 4.3.3, the RGD's central value is represented by the center of mass. The main drawback of this estimator is the fact that it is easily influenced by the presence of aberrant data [Bishop 2007, Afsari 2011, Formont *et al.* 2013]. To overcome this problem, we have recently introduced in [Hajri *et al.* 2016] a generative model for which the MLE of the central element is the Riemannian median. In other words, in order to enhance the model's robustness, the L_2 norm characterizing the RGD is replaced by the L_1 norm, giving the new distribution. This new model is called the Riemannian Laplace distribution and it is detailed next.

4.4.1 Definition

The development of this new distribution on the space $\mathcal{P}_m$ has been motivated by the need of a probabilistic model that is robust in the presence of outliers. Therefore, inspired from the well-known Laplace distribution on $\mathbb{R}$, we have introduced the Riemannian Laplace distribution (RLD) in [Hajri *et al.* 2016] on the space $\mathcal{P}_m$ of $m \times m$ real, symmetric and positive definite matrices. The probability density function of the RLD with respect to the Riemannian volume element is defined as:

$$p(\mathbf{M}|\bar{\mathbf{M}}, \sigma) = \frac{1}{\zeta(\sigma)} \exp\left\{ -\frac{d(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2} \right\}, \tag{4.34}$$

where $\bar{\mathbf{M}} \in \mathcal{P}_m$ and $\sigma > 0$ are the location and the dispersion parameters. $d(\,\cdot\,)$ is the Riemannian distance given in (4.5) and $\zeta(\sigma)$ is a normalization factor, independent of $\bar{\mathbf{M}}$. In the next section, more details on $\zeta(\sigma)$ are given.

4.4.2 Normalization Factor

4.4.2.1 Definition

By following the procedure previously introduced for the RGD, the normalization factor $\zeta(\sigma)$ is defined as [Hajri *et al.* 2016]:

$$\zeta(\sigma) = \int_{\mathcal{P}_m} \exp\left\{-\frac{d(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2}\right\} dv(\mathbf{M}), \tag{4.35}$$

with $dv(\mathbf{M})$ being the Riemannian volume element. Moreover, in the same work, it has been shown that $\zeta(\sigma)$ is independent of $\bar{\mathbf{M}}$, so by replacing it with the identity matrix, the following close form can be obtained:

$$\zeta(\sigma) = q_m \int_{\mathbb{R}^m} \exp\left\{-\frac{|\mathbf{r}|}{2\sigma^2}\right\} \prod_{i<j} \sinh\left\{\frac{|r_i - r_j|}{2}\right\} dr_1 \dots dr_m, \tag{4.36}$$

where $|\mathbf{r}| = (r_1^2 + \dots + r_m^2)^{\frac{1}{2}}$ and q_m is given in (4.13).

4.4.2.2 Numerical Computation of $\zeta(\sigma)$

In order to evaluate the expression in (4.36), the Monte Carlo integration can be used. Thus, the vector $\mathbf{r}$ has to be sampled from a multivariate Laplace distribution:

$$p(\mathbf{x}|\mathbf{M}) = \frac{1}{|\mathbf{M}|^{\frac{1}{2}}} \frac{\frac{1}{2}\,\Gamma_m(\frac{m}{2})}{\pi^{\frac{m}{2}}\,\Gamma_m(m)\,2^m} \exp\left\{-\frac{(\mathbf{x}^T\mathbf{M}^{-1}\mathbf{x})^{\frac{1}{2}}}{2}\right\}, \tag{4.37}$$

where $\Gamma_m(\cdot)$ is the multivariate Gamma function [Muirhead 1982] given in (4.14). Further on, $\mathbf{M}$ is considered to be σI_m:

$$p(\mathbf{x}|\sigma) = \frac{\frac{1}{2}\,\Gamma_m(\frac{m}{2})}{\pi^{\frac{m}{2}}\,\Gamma_m(m)\,2^p\,\sigma^{2m}} \exp\left\{-\frac{\sqrt{\mathbf{x}^T\mathbf{x}}}{2\sigma^2}\right\}. \tag{4.38}$$

In the end, the normalization factor in (4.36) becomes:

$$\zeta(\sigma) = \frac{\pi^{\frac{m}{2}}\,\Gamma_m(m)\,2^{(m+1)}\,\sigma^{2m}}{\Gamma_m(\frac{m}{2})} E\left[q_m \prod_{i<j} \sinh\left\{\frac{|x_i - x_j|}{2}\right\}\right], \tag{4.39}$$

where $E[\cdot]$ denotes the expectation with respect to the multivariate Laplace distribution defined in 4.38 and q_m has the form in (4.13). Similar to the RGD's normalization factor, the results are smoothed by means of cubic spline functions and then, tables containing the obtained values of $\zeta(\sigma)$ can be built.

In order to define the RLD, besides the normalization factor, two parameters are needed: the central value and the dispersion. Their estimation is the subject of the next part.

4.4.3 Parameter Estimation

Similar to the Riemannian Gaussian distribution, the parameters can be estimated through the maximum likelihood estimation.

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$ be a set of N independent and identically distributed samples according to a Riemannian Laplace distribution of central value $\bar{\mathbf{M}}$ and dispersion σ.

The MLE of the central value $\bar{\mathbf{M}}$ is the Riemannian median $\widehat{\mathbf{M}}$, obtained by minimizing the cost function:

$$f_{Med}(\bar{\mathbf{M}}) = \frac{1}{N} \sum_{n=1}^{N} d(\bar{\mathbf{M}}, \mathbf{M}_n), \tag{4.40}$$

where $d(\cdot)$ is the geodesic distance [James 1973] defined in (4.5). More details on the centroid estimation algorithm can be found in Section 5.2.2. With this definition for $\bar{\mathbf{M}}$, the construction of a robust parametric model is achieved. More precisely, the Riemannian median is more robust to outliers than the Riemannian center of mass [Fletcher *et al.* 2009] used in the case of RGDs. A detailed comparison between these two estimators is carried out in Chapter 5.

Further on, the MLE estimate $\hat{\sigma}$ of the dispersion σ is given by the solution of:

$$\sigma^3 \times \frac{d}{d\sigma}\zeta(\sigma) = f_{Med}(\widehat{\mathbf{M}}), \tag{4.41}$$

where $\zeta(\sigma)$ is the normalization factor. In practice, for $m = 2$, the dispersion $\hat{\sigma}$ is obtained by the Newton-Raphson algorithm [Hajri *et al.* 2016]. Like for the RGDs, if $m > 2$, then $\hat{\sigma}$ can be obtained by means of Monte Carlo integration. Thus, the derivative of $\zeta(\sigma)$ with respect to σ is expressed:

$$\begin{aligned} \frac{d}{d\sigma}\zeta(\sigma) &= \int_{\mathcal{P}_m} \frac{d(\mathbf{M}, \bar{\mathbf{M}})}{\sigma^3} \exp\left\{-\frac{d(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2}\right\} dv(\mathbf{M}) \\ &= q_m \times \int_{\mathbb{R}^m} \frac{|\mathbf{r}|}{\sigma^3} \exp\left\{-\frac{|\mathbf{r}|}{2\sigma^2}\right\} \prod_{i<j} \sinh\left\{\frac{|r_i - r_j|}{2}\right\} dr_1 \dots dr_m, \end{aligned} \tag{4.42}$$

where $|\mathbf{r}| = (r_1^2 + \ldots + r_m^2)^{\frac{1}{2}}$ and q_m is given in (4.13). Next, vector $\mathbf{r}$ is sampled from a multivariate Laplace distribution of parameter σI_m, resulting in:

$$\frac{d}{d\sigma}\zeta(\sigma) = \frac{\pi^{\frac{m}{2}}\ \Gamma(m)\ 2^{(m+1)}\ \sigma^{2m}}{\Gamma(\frac{m}{2})}\ E\left[q_m \frac{|\mathbf{x}|}{\sigma^3} \prod_{i<j} \sinh\left\{\frac{|x_i - x_j|}{2}\right\}\right]. \tag{4.43}$$

In addition, it has been shown that $\widehat{\mathbf{M}}$ and $\hat{\sigma}$ are unique and that $\widehat{\mathbf{M}}$ is a consistent estimator of $\bar{\mathbf{M}}$ [Hajri *et al.* 2016].

4.4.4 Mixture Model for RLDs

Starting from (4.34), the RLD definition has been extended to the case of mixtures of RLDs [Hajri *et al.* 2016]. For a mixture of K RLDs, the probability density function becomes:

$$p(\mathbf{M}|\theta) = \sum_{k=1}^{K} \varpi_k \, p(\mathbf{M}|\bar{\mathbf{M}}_k, \sigma_k), \tag{4.44}$$

where $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1 \leq k \leq K}\}$ is the parameter vector. ϖ_k are the positive weights, with $\varpi_k \in (0, 1)$ and $\sum_{k=1}^{K} \varpi_k = 1$, while $p(\mathbf{M}|\bar{\mathbf{M}}_k, \sigma_k)$ is given by (4.34).

The parameters of each component $k = 1, \ldots, K$ can be estimated by using the k-means, or the expectation maximization algorithms.

4.4.4.1 Parameters Estimation by Using the K-means Algorithm

The estimation procedure is similar to the one presented in Algorithm 1 for the RGDs. The two methods differ only in the definition of the update rules. More precisely, for robustness purpose, the parameters of each cluster c_k are computed by replacing the center of mass cost function in (4.22) and (4.23) by the median cost function:

$$f_{Med}(\bar{\mathbf{M}}_k) = \frac{1}{N_k} \sum_{n=1}^{N_k} d(\bar{\mathbf{M}}_k, \mathbf{M}_{k_n}), \tag{4.45}$$

with $\mathbf{M}_{k_n} \in c_k$, $n = 1, \ldots, N_k$ and N_k representing the cardinal of c_k. In addition, the normalization factor $Z(\sigma)$ in (4.23) is replaced by $\zeta(\sigma)$.

4.4.4.2 Parameters Estimation by Using the Expectation-Maximization Algorithm

The expectation-maximization algorithm has been also introduced for the Riemannian Laplace distributions [Hajri *et al.* 2016]. The general idea of the estimation method is similar to the one described in Algorithm 2 for the RGDs. In order to obtain the mixture's parameters, small changes have to be made in the previously introduced algorithm. First, for each mixture component k, $k = 1, \ldots, K$, the quantity $\omega_k(\mathbf{M}_n, \theta)$ in (4.25) has the following expression :

$$\omega_k(\mathbf{M}_n, \theta) = \frac{\varpi_k \times p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{s=1}^{K} \varpi_s \times p(\mathbf{M}_n|\bar{\mathbf{M}}_s, \sigma_s)}, \tag{4.46}$$

where $p(\cdot)$ represents the RLD probability density function. Second, the squared distances $d^2(\mathbf{M}, \mathbf{M}_n)$ in (4.27) and $d^2(\widehat{\bar{\mathbf{M}}}_k, \mathbf{M}_n)$ in (4.28) are replaced by $d(\mathbf{M}, \mathbf{M}_n)$ and $d(\widehat{\bar{\mathbf{M}}}_k, \mathbf{M}_n)$. Third, the estimated dispersion in (4.29) is defined by using Φ, which is the inverse function of $\sigma \mapsto \sigma^3 \times \frac{d}{d\sigma} \log \zeta(\sigma)$.

4.4.4.3 Bayesian Information Criterion for RLDs

For the Riemannian Laplace distribution, the number of mixture components K can be estimated by the BIC. The same idea as the one presented in Section 4.3.4.3 can be implemented, by simply replacing the probability density function in (4.32) with the RLD in (4.34).

4.5 Application to Texture Image Classification

In this section, the Riemannian mixture models are applied to texture image classification by using the MIT Vision Texture (VisTex) database [Vis]. The purpose of this experiment is to classify the textures, by taking into consideration the within-class diversity. Therefore, each texture class is characterized in the parameter space $\boldsymbol{\theta}$ by its central value $\bar{\mathbf{M}}$ and its dispersion σ, as illustrated in Figure 4.4. In this context, the influence of outliers on the classification performances is analyzed and the results are compared to those given by the Wishart distribution (WD) [Lee *et al.* 1999, Saint-Jean & Nielsen 2013].

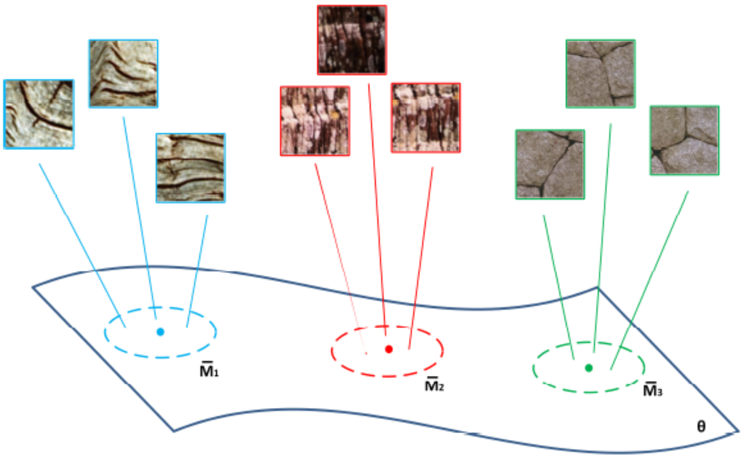
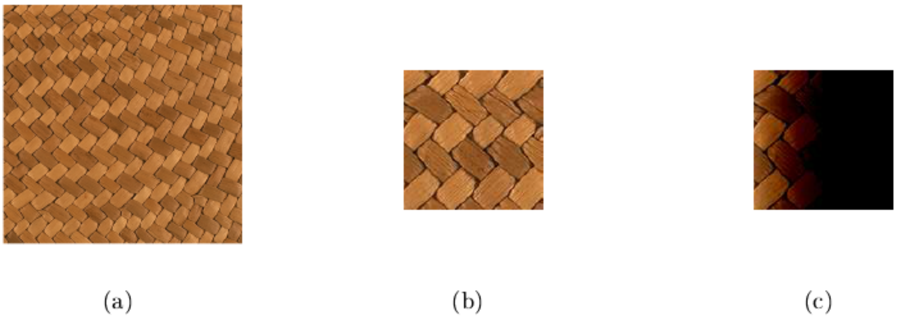


Figure 4.4: Within-class diversity modeled by Riemannian distributions.

4.5.1 Database

The VisTex database contains 40 images illustrated in Figure 2.1 and considered as being 40 different texture classes. Starting from this database, a modified version has been built, as follows. First, each texture is decomposed in 169 patches of 128×128 pixels, with an overlap of 32 pixels. As a result, a total number of 6760 textured patches are obtained. Next, some patches are corrupted, in order to introduce abnormal data into the dataset. Therefore, their intensity is modified by applying a gradient of luminosity. For each class, between 0 and 60 patches are modified in order to become outliers. An example of a VisTex texture with one of its patches and an outlier patch are shown in Figure 4.5.



(a) (b) (c)

Figure 4.5: (a) Example of a texture from the VisTex database, (b) one of its patches and (c) the corresponding outlier.

4.5.2 Methodology and Results

For this experiment, the classification is performed by using the EM algorithm. Knowing that this is a supervised classification algorithm, the database is 15 times equally and randomly divided in order to obtain the training and the testing sets. Then, for each patch in both databases, a feature vector has to be computed. The luminance channel is first extracted, and then normalised in intensity. The grayscale patches are filtered using the stationary wavelet transform with Daubechies db4 filter, with 2 scales and 3 orientations. Next, the wavelet coefficients located in a $p \times q$ spatial neighborhood of the current spatial position are clustered in a random vector and modeled as realisations of zero-mean multivariate Gaussian distributions. For this experiment, the spatial information is captured by using a vertical (2×1) and a horizontal (1×2) neighborhood. Further on, the 2×2 sample covariance matrices are estimated for each wavelet subband and each neighborhood. In the end, each patch is represented by a set of $F = 12$ covariance matrices (2 scales $\times$ 3 orientations $\times$ 2 neighborhoods).

The estimated covariance matrices are elements of $\mathcal{P}_m$, with $m = 2$ and therefore they can be modeled by Riemannian Gaussian distributions and Riemannian Laplace distributions. More precisely, in order to take into consideration the within-class diversity, each class c in the training set is viewed as a realisation of a mixture of Riemannian distributions with K mixture components, characterized by $\Theta_c = (\varpi_k^c, \bar{\mathbf{M}}_{k,f}^c, \sigma_{k,f}^c)$, having $\mathbf{M}_{k,f}^c \in \mathcal{P}_2$, with $k = 1, \cdots, K$ and $f = 1, \cdots, F$. Since the wavelet subbands are assumed independent, the probability density describing the training class c is:

$$p(\mathbf{M}|\Theta_c) = \sum_{k=1}^{K} \varpi_k^c \prod_{f=1}^{F} p(\mathbf{M}_f^c | \bar{\mathbf{M}}_{k,f}^c, \sigma_{k,f}^c), \tag{4.47}$$

where $p(\mathbf{M}_f^c | \bar{\mathbf{M}}_{k,f}^c, \sigma_{k,f}^c)$ is the Riemannian Gaussian distribution given in (4.5), or the Riemannian Laplace distribution in (4.34).

The learning step of the classification is performed using the EM algorithm for mixture models, presented earlier in this chapter. For the number of mixture components, several situations are considered. First, K is predefined and set to 1. In this case, the within-class diversity is only modeled by the dispersion around the centroids. Next, K is set to 3 and third, it is determined by using the BIC criterion recalled in (4.31). Note that for both RGD and RLD models, the degree of freedom is expressed as:

$$DF = K \times F \times \frac{m(m+1)}{2} + K \times F + (K-1), \tag{4.48}$$

since one centroid and one dispersion parameter should be estimated for each subband and for each component of the mixture model. In practice, the number of mixture components K varies between 2 and 5, and the K yielding to the lowest BIC criterion is retained.

The EM algorithm is sensitive to the initial conditions. In order to minimize this influence, for this experiment the EM algorithm is repeated 10 times and the result maximizing the log-likelihood functions is retained. Finally, the classification is performed by assigning each element $\mathbf{M} \in \mathcal{P}_2$ in the testing set to the class of the closest cluster c, maximizing one of the following log-likelihood criteria:

- for the mixture of K RGDs:

$$\arg\max_c \left\{ \log \hat{\varpi}_k^c - \sum_{f=1}^{F} \log Z(\hat{\sigma}_{k,f}^c) - \sum_{f=1}^{F} \frac{d^2(\mathbf{M}_{t,f}, \widehat{\mathbf{M}}_{k,f}^c)}{2(\hat{\sigma}_{k,f}^c)^2} \right\}; \tag{4.49}$$

- for the mixture of K RLDs:

$$\arg\max_c \left\{ \log \hat{\varpi}_k^c - \sum_{f=1}^{F} \log \zeta(\hat{\sigma}_{k,f}^c) - \sum_{f=1}^{F} \frac{d(\mathbf{M}_{t,f}, \widehat{\mathbf{M}}_{k,f}^c)}{2(\hat{\sigma}_{k,f}^c)^2} \right\}, \tag{4.50}$$

  where $\mathbf{M}_{t,f}$ is the sample covariance matrix of the f^{th} subband of the test patch t, knowing that F subbands are extracted for each patch.

It has to be mentioned that these two decision rules represent the extension of the quadratic discriminant analysis to the case when the image descriptors are covariance matrices.

In addition, the results are compared to the Wishart distribution (WD) [Lee *et al.* 1999, Saint-Jean & Nielsen 2013].

The classification results expressed in terms of overall accuracy are shown in Figure 4.6 for RGDs in black, RLDs in red and WD in blue. For all the considered methods, the classification rate is given as a function of the number of outliers, that varies between 0 and 60 for each class.

From this graphic, the influence of abnormal data on the RGD and RLD models is first analyzed as the number of outlier patches increases. The results show that the RLD gives slightly better results than the RGD. Next, the number of mixture components is considered. It can be noticed that the results are improved by using mixture distributions joint with the BIC criterion for choosing the suitable number

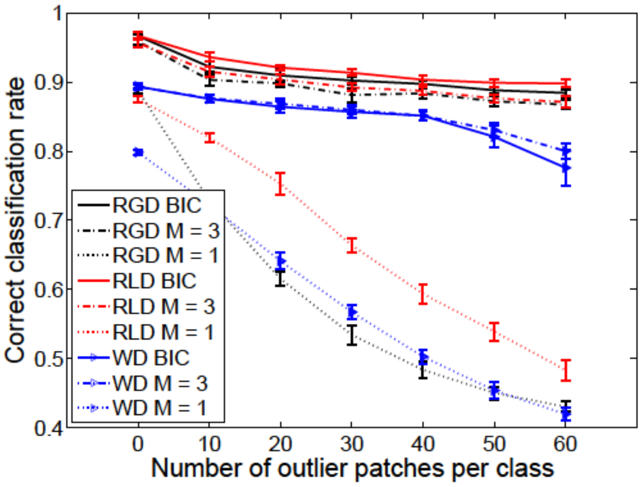


Figure 4.6: Classification results (Source: [Hajri *et al.* 2016].

of clusters. In conclusion, the mixture of RLDs combined with the BIC criterion to estimate the best number of mixtures components can minimize the influence of abnormal samples present in the dataset.

A second experiment is performed, having as purpose the comparison between the linear and quadratic discriminant analysis. In the following, the RLD BIC method is considered. The decision criterion for the quadratic discriminant analysis has been given in (4.50), while the linear discriminant analysis is obtained when the within-class diversity is not captured by means of the dispersion parameter. In other words, the homoscedasticity assumption is added to the mixture model in (4.50), meaning that all the clusters are characterized by the same dispersion $\hat{\sigma}^c_{k,f} = \sigma$. As a result, the following log-likelihood criterion has to be maximized:

$$\arg\max_{c} \left\{ -\sum_{f=1}^{F} d(\mathbf{M}_{t,f}, \widehat{\mathbf{M}}_f^c) \right\}, \tag{4.51}$$

which is also called the minimum distance to mean classifier [Barachant *et al.* 2012]. The classification results are presented in Figure 4.7, showing the importance of this parameter in the decision rule. For instance, if the number of outlier patches per class is fixed to 30, a significant gain of about 3.5% is observed when the quadratic discriminant analysis is considered.

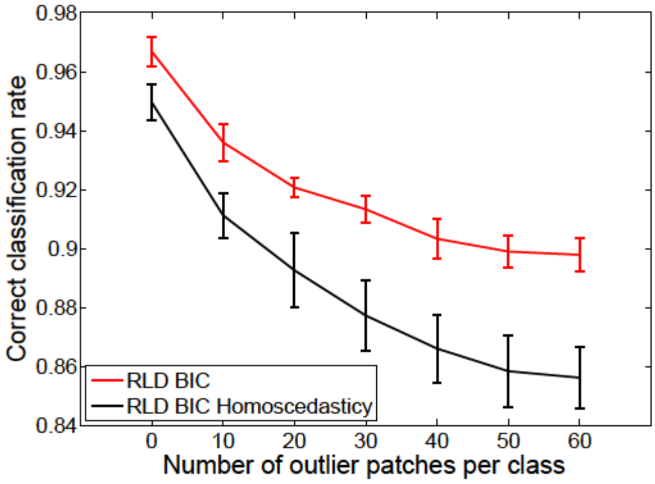


Figure 4.7: Comparison between the classification results given by the RLD BIC with and without the homoscedasticity assumption.

4.6 Conclusions and Perspectives

4.6.1 Conclusions

In this chapter, two probability density functions modeling the manifold of covariance matrices have been presented: the Riemannian Gaussian distribution and the Riemannian Laplace distribution. These models have been compared in the context of texture image classification by studying the influence of outliers on the classification results.

First, the Riemannian Gaussian distribution has been defined and details on the computation of its normalization factor have been given. This distribution is characterized by two parameters that are the central value and the dispersion. The main drawback of this probability model is the fact that its central value is given by the center of mass, which is a non-robust estimator. To solve this problem, we proposed the extension of the Laplace distribution to the Riemannian manifold, knowing that its central value is the median. The obtained density is called the Riemannian Laplace distribution.

Second, this new distribution has been presented, along with the computation scheme of its normalization factor.

Third, the parameter estimation process has been detailed for both RGD and RLD, by using the maximum likelihood estimation approach.

Next, the RGD and RLD have been extended to the mixture models and a modified version of the k-means and EM algorithms for the parameters' estimation has been proposed.

Moreover, the Bayesian information criterion has been adapted for the Riemannian manifold, in order to automatically compute the appropriate number of clusters.

In the end, both distribution models have been applied to texture image classification, by using a modified version of the VisTex database. The performed experiments have been design to study the influence of outliers on the two distributions and to analyze the importance of the dispersion in the construction of the decision rule. In addition, the classification results have been compared to those given by modeling the data using the Wishart distribution. The obtained results have shown that for a significant number of outliers, the correct classification performance is increased by considering the RLD model.

In this chapter, is has been shown that by using the median, the presence of outliers can be handled. Further on, in the following chapter, the centroid estimation methods will be analyzed and compared in terms of robustness to aberrant data. In addition, a new robust estimator will be introduced.

4.6.2 Perspectives

Future works will include:

- *The extension of the multivariate generalized Gaussian distribution to the Riemannian manifold:* in this chapter, two probability density functions defined on the Riemannian manifold have been presented: the Riemannian Gaussian distribution and the Riemannian Laplace distribution. The study of another function, that is the generalized Riemannian Gaussian distribution represents the subject of future works, along with the development of some appropriate centroid estimation methods. In this case, the probability density function will be of the form:

  $$p\left(\mathbf{M}|\bar{\mathbf{M}},\sigma\right) = \frac{1}{Z(\sigma)}\exp\left\{-\frac{d^{\beta}(\mathbf{M},\bar{\mathbf{M}})}{2\sigma^2}\right\}, \tag{4.52}$$

  where β is the shape parameter. In this case, the maximum likelihood estimator of the centroid $\bar{M}$ will be given by the p-means [Arnaudon *et al.* 2013]

- *The extension of the Riemannian distributions to the space of complex covariance matrices:* in the case of Polarimetric SAR data, complex covariance matrices are classically used as descriptors, in order to model the information contained in this type of images. Therefore, future works will address the problem of modeling this matrices on the Riemannian manifold, by extending the Riemannian distributions to the space of Hermitian positive definite matrices [Hajri *et al.* 2017].

- *The development of Riemannian models for structured covariance matrices:* in practice, covariance matrices having special forms can be encounterd in signal and image processing applications, like the Tœplitz, or block-Tœplitz matrices. For instance, the autocovariance matrices of wide-sense stationary random signals are Tœplitz matrices [Therrien 1992], while the block-Tœplitz matrices can be encountered in multi-channel linear prediction [Therrien 1981], or filtering problems [Jakobsson *et al.* 2000]. These structured matrices are characterized by specific properties that will be exploited in future works to develop models which take into account this particular geometry [Said *et al.* 2016].

Chapter 5

Robust Centroid Estimation on the Manifold of Covariance Matrices

Contents

5.1 Introduction

In the previous chapter, it has been shown that covariance matrices can be modeled as realizations of Riemannian Gaussian distributions or Riemannian Laplace distributions and used in classification algorithms such as k-means or Expectation-Maximization (EM) [Said *et al.* 2015a]. This kind of classification procedures are based on the partition of the dataset in subsets, or clusters, characterized by their central values, also called centroids. The dataset's partition is accomplished by assigning each observation to the closest cluster in terms of a predefined distance [Bishop 2007]. This is a recursive procedure and for each iteration, the centroid's value is recomputed and the assignation step is repeated. Often, the cluster's centroid is the center of mass, computed by using the squared Euclidean distance [MacQueen 1967, Lloyd 2006]. Despite its popularity, this method is not appropriate for covariance matrices having a Riemannian geometry. To solve this problem, the Euclidean distance can be replaced by an intrinsic metric such as the Riemannian distance. The obtained Riemannian center of mass has been defined in Chapter 4 and details on its computation will be given in the next section. However, the main disadvantage of the center of mass is its non-robust behavior to outliers that can exist in the dataset [Bishop 2007, Afsari 2011, Formont *et al.* 2013]. A robust alternative for the centroid's computation is the median, which has been also generalized for Riemannian manifolds [Yang 2010, Fletcher *et al.* 2009, Barbaresco *et al.* 2013]. In practice, the median is determined by using a gradient descent algorithm. Nonetheless, for its computation, a division by the distance between each observed covariance matrix in the dataset and the median is needed. If these two points are too close, this distance tends toward zero and may lead to numerical instability. In such case, Yang proposes to exclude those points, at each iteration of the algorithm [Yang 2010]. Another possibility for determining robust centroids in the space of covariance matrices is the use of the trimming methods [Uehara *et al.* 2016]. These algorithms imply the elimination of a fixed percentage of outliers, according to their distance with respect to the dataset's mean or median, and the computation of the mean or the median on the remaining data. Nevertheless, the main difficulty of the trimmed estimators relies on how the percentage of discarded data can be tuned.

In this chapter, a novel centroid estimator, based on the theory of M-estimators is proposed. By considering the so-called Huber's function [Huber 1964, Tyler 1987], the definition of this estimator is introduced and an algorithm to estimate it from a sample of N covariance matrices is presented. The proposed estimator is a trade-off between the center of mass and the median, where the former is efficient, while the latter is robust to outliers. Moreover, a method to automatically determine the Huber's threshold is presented, based on the median absolute deviation (MAD) concept [Ilea *et al.* 2016c, Ilea *et al.* 2016d].

The chapter is structured as follows. Section 5.2 recalls the definition of the centroid of a sample of N observations. An overview of the center of mass, the median and the trimming based methods is also given. Section 5.3 introduces the proposed Huber's centroid estimator and presents a gradient descent algorithm to estimate it. In addition, an algorithm to automatically tune the Huber's threshold is developed. Section 5.4 evaluates the performance of these estimators on simulated data. In Section 5.5, the results are validated through two applications concerning the texture image classification and the brain decoding. Section 5.6 draws a parallel between the robust estimators of covariance matrices and the robust estimators of centroids. Their importance in the classification workflow is illustrated in several examples. In the end, Section 5.7 reports some conclusions and perspectives.

5.2 Centroids and Estimation Methods

Let $\mathcal{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ be a random sample of N covariance matrices, characterized by its central value $\bar{\mathbf{M}}$. The estimated centroid of this set, denoted $\widehat{\bar{\mathbf{M}}}$, is obtained by minimizing the following cost function $f(\bar{\mathbf{M}})$:

$$\widehat{\bar{\mathbf{M}}} = \arg\min_{\bar{\mathbf{M}}} f(\bar{\mathbf{M}}). \tag{5.1}$$

In practice, the minimum value of $f(\bar{\mathbf{M}})$ is found by using gradient based algorithms [Absil *et al.* 2008]. Thus, the centroid is recursively estimated by using the following expression:

$$\bar{\mathbf{M}}_{it+1} = \mathrm{Exp}_{\bar{\mathbf{M}}_{it}}(-s_{it}\ \nabla f(\bar{\mathbf{M}}_{it})), \tag{5.2}$$

with s_{it} being the descent step and $\mathrm{Exp}_{\mathbf{M}}(\cdot)$ the Riemannian exponential mapping [Higham 2008] given in (4.6). Moreover, the Armijo's backtracking procedure [Armijo 1966] is used to fix s_{it} at each iteration it. This recursive process is repeated as long as the norm of $\nabla f(\bar{\mathbf{M}}_{it})$, denoted D_{it}, is greater than a precision parameter ε, or until a maximum number of iterations N_{max} is reached. More precisely, D_{it} is given by:

$$D_{it} = ||\nabla f(\bar{\mathbf{M}}_{it})|| = \mathrm{tr}\left((\bar{\mathbf{M}}_{it}^{-1} \nabla f(\bar{\mathbf{M}}_{it}))^2\right). \tag{5.3}$$

Depending on the choice of $f(\bar{\mathbf{M}})$, different centroid estimators have been introduced in the literature. In this section, the definition of two well-known estimators, that are the center of mass [Karcher 1977,Nielsen & Bhatia 2012,Fiori 2009] and the median [Fletcher *et al.* 2009,Yang *et al.* 2010] are recalled. In addition, the methods based on the geometric trimmed averages [Uehara *et al.* 2016] are presented.

5.2.1 The Center of Mass

The center of mass (CM) has been first introduced in [Karcher 1977] and it became one of the most popular estimators. In this case, the estimated centroid is obtained by minimizing the sum of squared distances between the centroid $\bar{\mathbf{M}}$ and the observations $\mathbf{M}_i$, $i = 1, \ldots, N$. Therefore, the cost function is:

$$f_{CM}(\bar{\mathbf{M}}) = \frac{1}{N} \sum_{i=1}^{N} d^2(\bar{\mathbf{M}}, \mathbf{M}_i), \tag{5.4}$$

where $d(\cdot)$ represents the Rao's Riemannian distance between two covariance matrices introduced in (4.5) and defined as [James 1973]:

$$d(\mathbf{M_1}, \mathbf{M_2}) = \left[\sum_{i=1}^{p} (\ln \lambda_i)^2 \right]^{\frac{1}{2}}, \tag{5.5}$$

where λ_i, $i = 1 \dots m$ are the eigenvalues of $\mathbf{M}_2^{-1}\mathbf{M}_1$ and m is the size of covariance matrices. In the same paper, a gradient-based algorithm has been proposed for the computation of the center of mass. Starting from (5.4), the gradient with respect to $\bar{\mathbf{M}}$, denoted by $\nabla f_{CM}(\bar{\mathbf{M}})$, is defined as:

$$\nabla f_{CM}(\bar{\mathbf{M}}) = -\frac{2}{N} \sum_{i=1}^{N} \mathrm{Log}_{\bar{\mathbf{M}}}(\mathbf{M}_i), \tag{5.6}$$

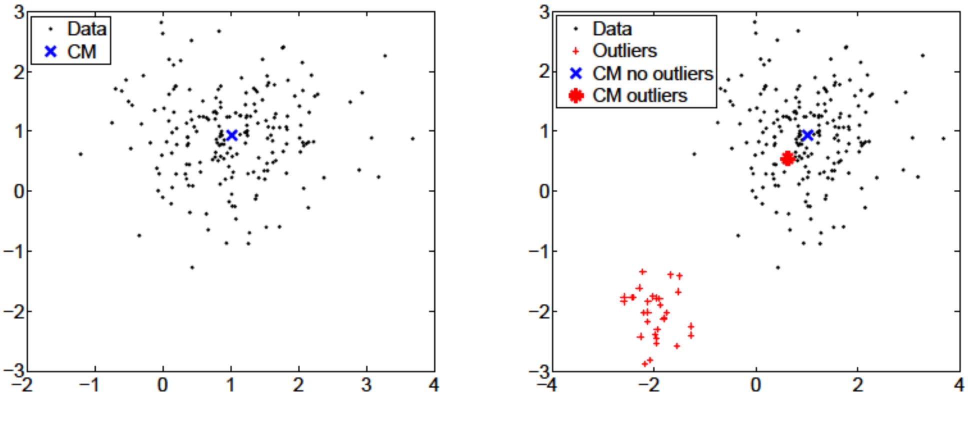
where $\mathrm{Log}_{\bar{\mathbf{M}}}(\cdot)$ is the Riemannian logarithm mapping [Higham 2008] given in (4.7). This function is used further, to recursively estimate the centroid. A pseudo-code describing this procedure is presented in Algorithm 3, knowing that $D_{CM_{it}}$ represents the gradient's norm obtained from (5.3) and (5.6).

Algorithm 3 Center of mass estimator

```
Input: $\mathbf{M_1}, \dots, \mathbf{M}_N$, $\varepsilon$, $N_{\max}$
Initialize $\bar{\mathbf{M}}$ using the sample mean
$it = 1$
while ($D_{CM_{it}} > \varepsilon$) and ($it \le N_{\max}$) do
    Estimate $\bar{\mathbf{M}}$ using one iteration of (5.2).
    Compute the gradient norm, $D_{CM_{it}}$, according to (5.3).
    $it = it + 1$.
end while
Output: $\bar{\mathbf{M}}$
```

The center of mass has been also studied in works like [Moakher 2006, Pennec 2006, Nielsen & Bhatia 2012, Fiori 2009]. Even though it is largely used, this method has a major drawback: it is easily influenced by the outliers present in the dataset [Yang 2010, Fletcher *et al.* 2009]. This idea is illustrated by an example in Figure 5.1. First, the CM for an outlier-free dataset is computed (Figure 5.1.a). Next, outliers are added and the CM is recomputed (Figure 5.1.b). It can be seen that, in this case, the centroid is attracted by the aberrant data, which proves its non robust behavior.

In order to reduce the impact of aberrant data on the estimated centroid's value, several possibilities are available. Some authors have proposed in [Fletcher *et al.* 2009, Uehara *et al.* 2016] the use of trimming based methods to remove the outliers before the computation of (5.4), or the use of other estimators generalized for the Riemannian space, like the median [Fletcher *et al.* 2009, Yang 2010].



(a) (b)

Figure 5.1: Behavior of the center of mass for (a) an outlier-free dataset and (b) in the presence of outliers.

5.2.2 The Median

The median is a robust centroid estimator, computed by minimizing the sum of distances between the centroid $\bar{\mathbf{M}}$ and the observations $\mathbf{M}_i$. Thus, the cost function is:

$$f_{Med}(\bar{\mathbf{M}}) = \frac{1}{N} \sum_{i=1}^{N} d(\bar{\mathbf{M}}, \mathbf{M}_i), \tag{5.7}$$

where $d(\,\cdot\,)$ is the Riemannian distance, given in (5.5).

This estimator has been first generalized to the Riemannian case in [Fletcher *et al.* 2009]. In their work, the authors considered the cost function as being the weighted sum of distances. Note that the equation in (5.7) is obtained when all the weights are equal to $1/N$. In order to compute the median's value, they have proposed to extend the Weiszfeld [Weiszfeld 1937] algorithm to manifolds. More precisely, the median is iteratively updated by using a subgradient algorithm on the cost function. In addition, they proved the algorithm's convergence to a unique value for positively curved manifolds. In [Yang 2010, Yang *et al.* 2010], the authors have defined the Riemannian median for the complete Riemannian manifold and they have introduced a gradient-based estimation algorithm that converges for both positively and negatively curved manifolds. In the following, the gradient descent estimation algorithm is detailed, starting from the cost function given in (5.7). In this case, the gradient with respect to $\bar{\mathbf{M}}$, denoted by $\nabla f_{Med}(\bar{\mathbf{M}})$, can be written as:

$$\nabla f_{Med}(\bar{\mathbf{M}}) = -\frac{1}{N} \sum_{i=1}^{N} \frac{\mathrm{Log}_{\bar{\mathbf{M}}}(\mathbf{M}_i)}{d(\bar{\mathbf{M}}, \mathbf{M}_i)}, \tag{5.8}$$

where $\mathrm{Log}_{\bar{\mathbf{M}}(\,\cdot\,)}$ is the Riemannian logarithm mapping [Higham 2008] given in (4.7).

As it can be seen, the gradient in (5.8) exists only if $d(\bar{\mathbf{M}}, \mathbf{M}_i) \neq \mathbf{0}$. In some cases, that is when observations $\mathbf{M}_i$, $i = 1, \ldots, N$ are too close to the current centroid's estimate $\bar{\mathbf{M}}_{it}$, the distance between them is close to $\mathbf{0}$, yielding potential numerical instability. To avoid this situations, in [Yang 2010] the author proposed to exclude, at each iteration it, the observations $\mathbf{M}_i$ that are too close from $\bar{\mathbf{M}}_{it}$. Therefore, a threshold value T is needed to define the proximity between the estimated centroid and the observations. More precisely, if

$$d(\bar{\mathbf{M}}_{it}, \mathbf{M}_i) \leq T, \tag{5.9}$$

then the observations $\mathbf{M}_i$ are discarded at iteration it. Next, for iteration, $it+1$, the previously discarded observations are reintroduced into the dataset and the estimation and distance verification steps are repeated. This process is iterated for a fixed number of iterations N_{max}, or until the gradient's norm $D_{Med_{it}}$ is smaller than the predefined value ε. A pseudo-code is given in Algorithm 4, in order to synthesize the median centroid estimation, knowing that $D_{Med_{it}}$ is obtained from (5.3) and (5.8).

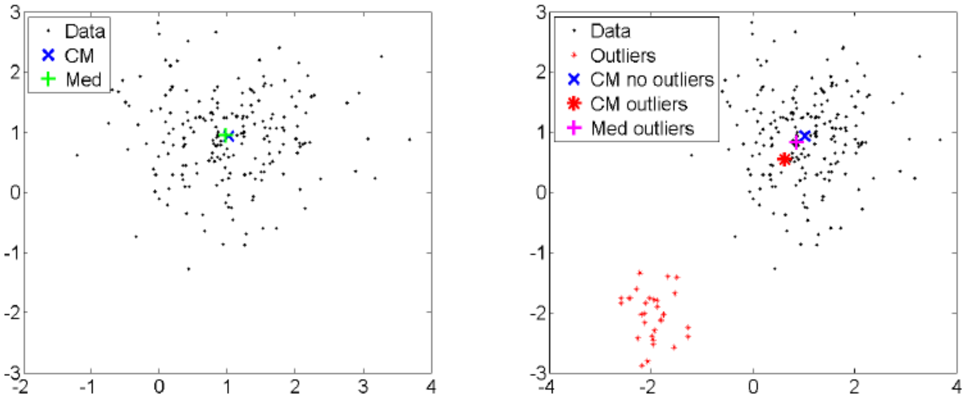
Algorithm 4 Median centroid estimator

```
Input: $\mathbf{M}_1, \ldots, \mathbf{M}_N$, $T$, $\varepsilon$, $N_{\max}$
Initialize $\bar{\mathbf{M}}$ using the sample mean.
$it = 1$.
while $(D_{Med_{it}} > \varepsilon)$ and $(it \leq N_{\max})$ do
    Compute $d(\bar{\mathbf{M}}, \mathbf{M}_i)$.
    for $i = 1, \ldots, N$ do
        if $d(\bar{\mathbf{M}}, \mathbf{M}_i) \leq T$ then
            Discard $\mathbf{M}_i$.
        end if
    end for
    Estimate $\bar{\mathbf{M}}$ using one iteration of (5.2).
    Compute the gradient norm, $D_{Med_{it}}$, according to (5.3).
    Reintroduce all the discarded $\mathbf{M}_i$.
    $it = it + 1$.
end while
Output: $\bar{\mathbf{M}}$
```

In Figure 5.2, the center of mass and the median are compared in terms of robustness to outliers. First, a dataset with no aberrant data is considered. In this case, the two estimators give almost identical centroids (Figure 5.2.a). On the other hand, when outliers are added (Figure 5.2.b), it can be noticed that the center of mass moves towards them, while the estimated median stays closer to the dataset's real central value.

5.2.3 The Geometric Trimmed Averages

The geometric trimmed averages [Uehara *et al.* 2016] are approaches that deal with outliers by eliminating them from the dataset. These methods are based on one



(a) (b)

Figure 5.2: Comparison between the center of mass and the median for (a) an outlier-free dataset and (b) in the presence of outliers.

parameter denoted α which represents the proportion of ignored data. Usually, $\alpha\%$ of the farthest observations are discarded.

For implementing these methods, several steps are needed. First, the centroid $\bar{\mathbf{M}}_0$ of the original dataset is obtained, by using the center of mass, or the median. Second, the Riemannian distances $V(i) = d(\bar{\mathbf{M}}_0, \mathbf{M}_i)$, $i = 1, \dots, N$ between the estimated centroid and the dataset's elements are computed. Next, $\alpha\%$ of the farthest elements from the estimated centroid $\bar{\mathbf{M}}_0$ are discarded. In the end, the center of mass, or the median of the remaining elements is recomputed.

Based on the centroid's estimation method, the following algorithms have been proposed in [Uehara *et al.* 2016]:

a) *geometric trimmed means*: the centroids are obtained by minimizing one of the following cost functions:

$$f_{CM_{Tmean}}(\bar{\mathbf{M}}) = f_{CM}(\mathrm{Trim}_{\alpha}^{\mathrm{mean}}(\mathbf{M})), \tag{5.10}$$

or

$$f_{CM_{Tmed}}(\bar{\mathbf{M}}) = f_{CM}(\mathrm{Trim}_{\alpha}^{\mathrm{med}}(\mathbf{M})), \tag{5.11}$$

where $f_{CM}(\cdot)$ is the center of mass cost function in (5.4), $\mathrm{Trim}_{\alpha}^{\mathrm{mean}}(\cdot)$ and $\mathrm{Trim}_{\alpha}^{\mathrm{med}}(\cdot)$ are the trimming operators, when $\alpha\%$ of the farthest elements from the center of mass, respectively the median are discarded;

b) *geometric trimmed medians*: the centroids are obtained by minimizing one of the following cost functions:

$$f_{Med_{Tmean}}(\bar{\mathbf{M}}) = f_{Med}(\mathrm{Trim}_{\alpha}^{\mathrm{mean}}(\mathbf{M})), \tag{5.12}$$

or

$$f_{Med_{Tmed}}(\bar{\mathbf{M}}) = f_{Med}(\mathrm{Trim}_{\alpha}^{\mathrm{med}}(\mathbf{M})), \tag{5.13}$$

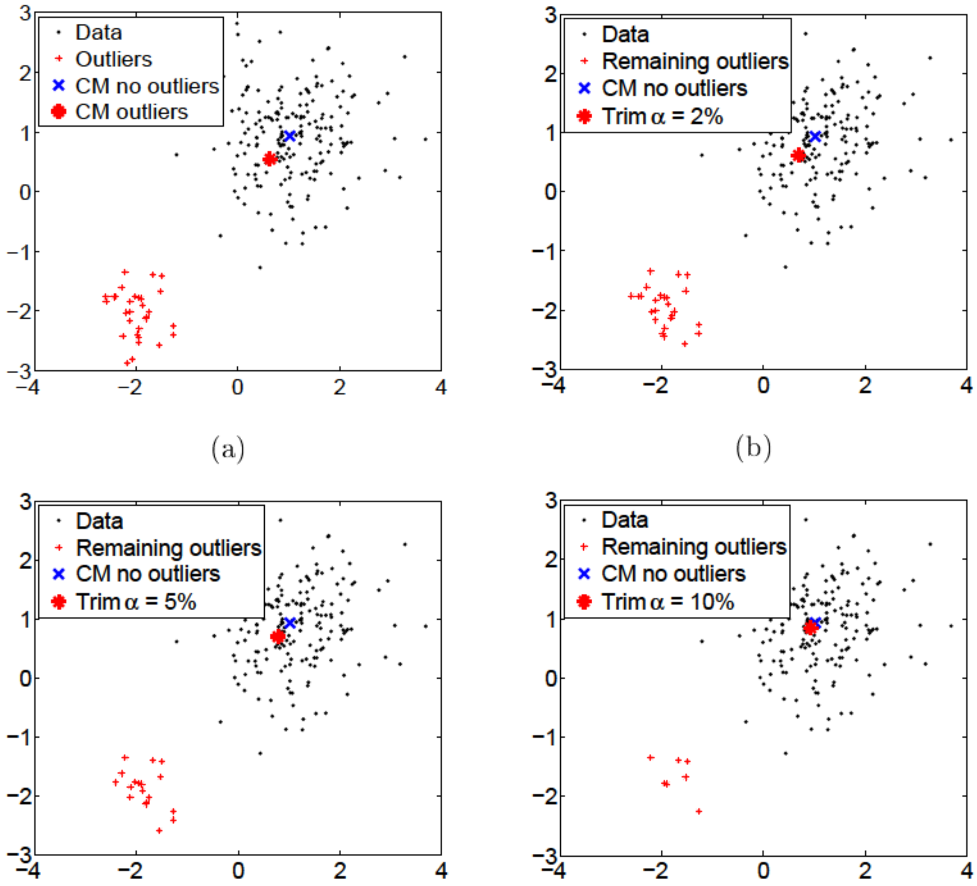
where $f_{Med}(\cdot)$ is the median cost function in (5.7).

To summarize all these methods, a pseudo-code is given in Algorithm 5.

Algorithm 5 Geometric Trimmed Averages

```
Input: $\mathbf{M}_1, \ldots, \mathbf{M}_N$, $\alpha$
Initialize $\bar{\mathbf{M}}_0$ using the center of mass, or the median.
Compute $\mathbf{V}(i) = d(\bar{\mathbf{M}}_0, \mathbf{M}_i)$, $i = 1, \ldots, N$ according to (5.5).
Discard $\alpha\%$ of the largest values in $\mathbf{V}$.
Compute $\bar{\mathbf{M}}$ using the center of mass, or the median.
Output: $\bar{\mathbf{M}}$
```

In Figure 5.3, the behavior of the geometric trimmed mean defined in (5.10) is shown, for different values of α. The original dataset is shown in Figure 5.3.a, where the red crosses represent the outliers. Starting from this dataset, $\alpha = 2\%$, 5% and 10% of the farthest elements from the centroid are discarded, giving Figure 5.3.b,



(a)

(b)

(c)

(d)

Figure 5.3: Behavior of the geometric trimmed mean defined in (5.10) for (a) $\alpha = 0\%$, which is equivalent to the CM of the entire set, (b) $\alpha = 2\%$, (c) $\alpha = 5\%$ and (d) $\alpha = 10\%$.

Figure 5.3.c and Figure 5.3.d. In these three last figures, the red crosses are the outliers remaining after the trimming procedure.

By analyzing the images, it can be noticed that as α increases, the estimated centroid becomes closer to its true value.

5.3 The Huber's Estimator

5.3.1 Motivation

In the previous section, several centroid estimators have been presented. These methods have been recently studied for covariance matrices in the Riemannian space [Fletcher *et al.* 2009, Yang 2010, Barbaresco *et al.* 2013, Uehara *et al.* 2016]. In the following, some disadvantages of these approaches are identified:

- *The center of mass* is known as being easily influenced by aberrant data.

- *The median* computation may lead to problems of numerical instability. The gradient of the cost function in (5.8) implies the division by the distance $d(\bar{\mathbf{M}}, \mathbf{M}_i)$ between the centroid $\bar{\mathbf{M}}$ and the observations $\mathbf{M}_i$. As mentioned earlier in Section 5.2.2, there are cases when this distance can be equal to zero. Therefore, the gradient function is not defined. In order to avoid these situations a threshold value has to be tuned for eliminating the observations that are too close from the estimated centroid. The problem that arises with this approach concerns the user dependent choice of the threshold's value.

- *The geometric trimmed averages* discard the outliers. Nevertheless, by deleting the elements that differ from the rest of the dataset, some new ones might become outliers. If the removal procedure is repeated, the dataset may become too small for further reliable analysis [Fletcher *et al.* 2009]. Moreover, another difficulty encountered with trimming based methods concerns the choice of α, the parameter fixing the percentage of eliminated observations.

In this context, to circumvent all those drawbacks, a novel centroid estimator on the manifold of covariance matrices is defined. The proposed method, called the Huber's estimator, can be viewed as a trade-off between the center of mass and the median, where the former is efficient, while the latter is robust to outliers. The compromise between these two estimators can be controlled by one parameter, the Huber's threshold. Its value can be automatically fixed, by taking into consideration the variability presented in the dataset.

5.3.2 Definition

In the following, the novel centroid estimator is introduced, based on the theory of M-estimators [Huber 1964, Maronna 1976, Tyler 1987]. In this case, the cost function in (5.1), denoted $f_u(\bar{\mathbf{M}})$ for the M-estimator, can be expressed by means of a scalar weight function $u(\cdot)$, as follows:

$$f_u(\bar{\mathbf{M}}) = \frac{1}{N}\sum_{i=1}^{N} u\big(d(\bar{\mathbf{M}}, \mathbf{M}_i)\big) d^2(\bar{\mathbf{M}}, \mathbf{M}_i), \tag{5.14}$$

where $u(\cdot)$ is a positive-valued function which gives a weight to each observation $\mathbf{M}_i$ in the computation of the centroid. Obviously, the weight function $u(\cdot)$ should decrease to zero to ensure that the outliers have a smaller contribution to the centroid's estimation than the other observations. Note that even if the center of mass (5.4) and the median (5.7) have expressions similar to (5.14) for respectively $u(d(\bar{\mathbf{M}}, \mathbf{M}_i)) = 1$ and $u(d(\bar{\mathbf{M}}, \mathbf{M}_i)) = \frac{1}{d(\bar{\mathbf{M}}, \mathbf{M}_i)}$, they do not belong to the family of M-estimators, since the regularity conditions of their corresponding weight function $u(\cdot)$ defined in [Maronna 1976] are not satisfied. These conditions have been mentioned in Section 3.2.4 in the context of covariance matrix estimation and they are explained next for centroid estimation:

- for the median: the weight function $u(\cdot)$ is not defined when $d(\bar{\mathbf{M}}, \mathbf{M}_i) = 0$;
- for the center of mass: the upper limit of $\psi(d(\bar{\mathbf{M}}, \mathbf{M}_i)) = d(\bar{\mathbf{M}}, \mathbf{M}_i)u(d(\bar{\mathbf{M}}, \mathbf{M}_i)) = d(\bar{\mathbf{M}}, \mathbf{M}_i)$ is infinite.

In [Huber 1964], Huber has introduced the so-called Huber's function $u(\cdot)$ defined as:

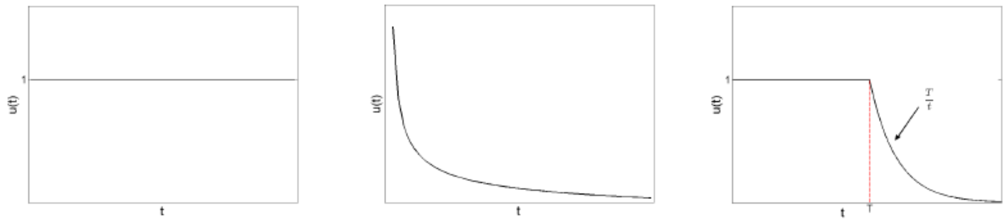
$$u\big(d(\bar{\mathbf{M}}, \mathbf{M}_i)\big) = \min\Big(1, \frac{T}{d(\bar{\mathbf{M}}, \mathbf{M}_i)}\Big), \tag{5.15}$$

where T is a threshold value controlling the contribution of outliers in the estimation. By combining (5.14) and (5.15), the proposed Huber's centroid is the covariance matrix $\bar{\mathbf{M}}$, which minimizes the following cost function [Ilea *et al.* 2016c, Ilea *et al.* 2016d]:

$$\begin{aligned} f_H(\bar{\mathbf{M}}) = &\frac{1}{N}\sum_{i=1}^{N} d^2(\bar{\mathbf{M}}, \mathbf{M}_i)\ \mathbb{1}_{\{d(\bar{\mathbf{M}}, \mathbf{M}_i)\leq T\}} \\ &+ \frac{T}{N}\sum_{i=1}^{N} d(\bar{\mathbf{M}}, \mathbf{M}_i)\ \mathbb{1}_{\{d(\bar{\mathbf{M}}, \mathbf{M}_i\}>T\}}, \end{aligned} \tag{5.16}$$

where $\mathbb{1}_{\{a\leq b\}}$ is the indicator function, which equals 1 if $a \leq b$ and 0 otherwise. The threshold T represents a measure of discriminating between normal and aberrant data and therefore, it controls the estimator's behavior. In other words, for large values of T, the Huber's estimator behaves as the center of mass, while for small values it is equivalent to the median. Figure 5.4 presents the Huber's function for the center of mass (Figure 5.4.a), the median (Figure 5.4.b) and the Huber's estimator (Figure 5.4.c).

In the following, a computation algorithm for the Huber's centroid is proposed, based on the gradient descent algorithm which minimizes the distance function given



(a) (b) (c)

Figure 5.4: The Huber's function $u(t)$ for (a) the center of mass, (b) the median and (c) the Huber's centroid.

in (5.16). The gradient of $f_H(\bar{\mathbf{M}})$ with respect to $\bar{\mathbf{M}}$ that is $\nabla f_H(\bar{\mathbf{M}})$ can be written as:

$$\nabla f_H(\bar{\mathbf{M}}) = -\frac{2}{N}\sum_{i=1}^{N} \mathrm{Log}_{\bar{\mathbf{M}}}(\mathbf{M}_i)\ \mathbb{1}_{\{d(\bar{\mathbf{M}},\mathbf{M}_i)\leq T\}} \\ -\frac{T}{N}\sum_{i=1}^{N} \frac{\mathrm{Log}_{\bar{\mathbf{M}}}(\mathbf{M}_i)}{d(\bar{\mathbf{M}},\mathbf{M}_i)}\ \mathbb{1}_{\{d(\bar{\mathbf{M}},\mathbf{M}_i\}>T\}}, \tag{5.17}$$

where $\mathrm{Log}_{\bar{\mathbf{M}}}(\,\cdot\,)$ is the Riemannian logarithm mapping [Higham 2008] given in (4.7). Once that this function is defined, it is further used in the recursive estimation procedure described by (5.2). A pseudo-code description of the Huber's centroid estimation is given in Algorithm 6, where $D_{H_{it}}$ is the gradient's norm obtained from (5.3) and (5.17).

Algorithm 6 Huber's centroid estimator

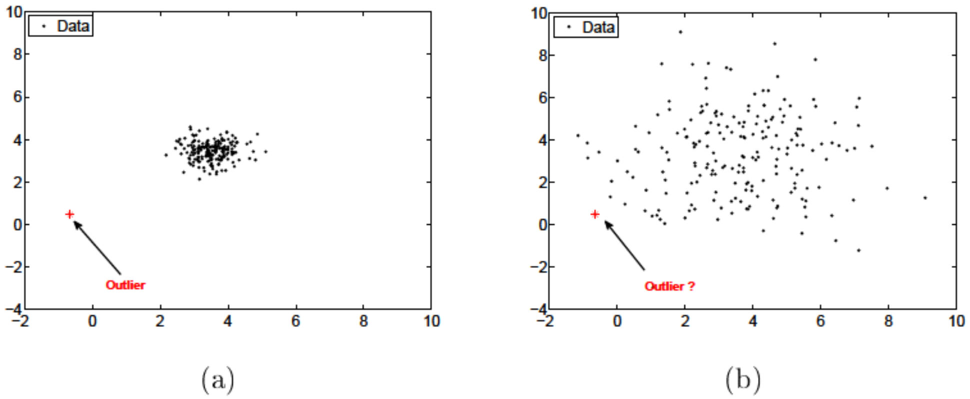
```
Input: $\mathbf{M}_1, \ldots, \mathbf{M}_N$, $T$, $\varepsilon$, $N_{\max}$
Initialize $\bar{\mathbf{M}}$ using the sample mean
$it = 1$
while ($D_{H_{it}} > \varepsilon$) and ($it \leq N_{\max}$) do
    Estimate $\bar{\mathbf{M}}$ using one iteration of (5.2).
    Compute the gradient norm, $D_{H_{it}}$, according to (5.3).
    $it = it + 1$
end while
Output: $\bar{\mathbf{M}}$
```

As observed in (5.17), the first and second terms correspond to the gradient of the cost function for the center of mass (5.6) and the median (5.8) centroids. For the second term, it can be seen that the division by distance $d(\bar{\mathbf{M}}_{it}, \mathbf{M}_i)$ is needed. As mentioned earlier, for the median, this division may cause computational problems. By using the proposed Huber's centroid, this problem is solved automatically by considering the threshold T. In conclusion, by choosing an appropriate value for T, the division by zero in the gradient function (5.17) will be avoided, which represents an important advantage of the proposed method.

In the following section, a user independent method is proposed in order to tune this parameter.

5.3.3 Algorithm for Huber's Threshold Automatic Computation

Similar to the geometric trimmed averages, the performance of the Huber's estimator depends greatly on the threshold T that discriminates between aberrant and normal data. Therefore the need to automatically fix it or at least to give an idea on its order of magnitude. In practice, T is application dependent and it is related to the intrinsic variability of the observed data. A visual explanation of this remark is given in Figure 5.5. For instance, if a dataset is characterized by a low variability, the outliers can be easily spotted (Figure 5.5.a). On the other hand, once that the dataset's variability increases, the outliers are much more difficult to identify (Figure 5.5.b). As T has to discriminate between outliers and normal data, its value should take into consideration this aspect.



(a)

(b)

Figure 5.5: Outliers and data intrinsic variability.

In the following, the data is considered to be a set of N covariance matrices $\mathcal{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ of size $m \times m$ distributed according to a Riemannian Gaussian distribution (RGD), detailed in Chapter 4. This distribution is characterized by two parameters: the central value $\bar{\mathbf{M}}$ and the dispersion σ. Its probability density function with respect to the Riemannian volume element is given by:

$$p(\mathbf{M}|\bar{\mathbf{M}}, \sigma) = \frac{1}{Z(\sigma)} \exp\Big\{ -\frac{d^2(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2} \Big\}, \tag{5.18}$$

where $Z(\sigma)$ is a normalization factor independent of the centroid $\bar{\mathbf{M}}$, and $d(\mathbf{M}, \bar{\mathbf{M}})$ is the Riemannian distance defined in (5.5).

In order to estimate the threshold's value, a robust estimator of the dispersion parameter σ is required, considering that the Huber's threshold T can be computed as:

$$T = c \times \hat{\sigma}, \tag{5.19}$$

with c being a constant and $\hat{\sigma}$ the estimated dispersion. In practice, c can take values between 1 and 2 [Huber & Ronchetti 2009]. As mentioned in [Huber & Ronchetti 2009], a common value is $c = 1.5$. For the dispersion parameter estimation, a robust method is introduced next. Inspired by the previous works on robust statistics [Huber & Ronchetti 2009], the concept of median absolute deviation (MAD) is extended to the case of covariance matrices living in the Riemannian space [Ilea *et al.* 2016c]. The MAD of $\mathcal{M}$ is defined as the median of the Riemannian distances d computed between each sample $\mathbf{M}_i$, $i = 1, \dots, N$ and the Riemannian median, denoted $RMed(\mathcal{M})$:

$$\text{MAD} = \text{median}\left(d\big(\mathbf{M}_i, RMed(\mathcal{M})\big)\right). \tag{5.20}$$

A comparison between the MAD in the Euclidean and the Riemannian spaces is made in Table 5.1.

Table 5.1: Definition of MAD in both Euclidean and Riemannian spaces.

Euclidean Space	Riemannian Manifold
Let $\mathbf{X} = \{X_1, \dots, X_N\}$ be a set of scalar observations: $\text{MAD} = \text{median}(\lvert X_i - \text{median}(\mathbf{X})\rvert)$.	Let $\mathcal{M} = \{\mathbf{M}_1, \dots, \mathbf{M}_N\}$ be a set of covariance matrices: $\text{MAD} = \text{median}\left(d\big(\mathbf{M}_i, RMed(\mathcal{M})\big)\right)$.

Further on, a link between the MAD and the dispersion parameter σ is needed. More precisely, the MAD is defined as:

$$\frac{1}{2} = p\big(d(\mathbf{M}, \bar{\mathbf{M}}) \leq \text{MAD}\,\big) = p\left(\frac{d(\mathbf{M}, \bar{\mathbf{M}})}{m\sigma} \leq \frac{\text{MAD}}{m\sigma}\right). \tag{5.21}$$

Starting from this expression, a new variable is introduced:

$$z = \frac{d(\mathbf{M}, \bar{\mathbf{M}})}{m\sigma}, \tag{5.22}$$

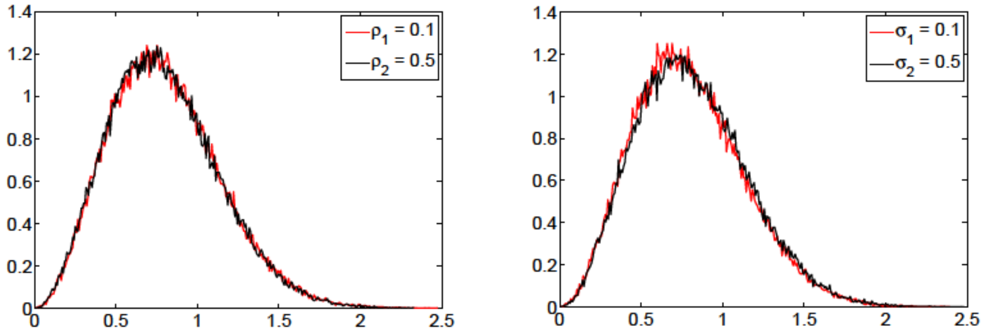
and its statistics are studied. In practice, it has been observed on simulated data that the distribution of z is independent of $\bar{\mathbf{M}}$ and σ. To sustain this remark, an example is shown in Figure 5.6. The behavior of z has been analyzed in the following experiment. A dataset of 10^5 independent and identically distributed covariance matrices of size $m \times m$, issued from an RGD model has been generated. The simulated covariance matrix dataset has been obtained for centroids $\bar{\mathbf{M}}$ having the form:

$$\bar{\mathbf{M}}(i,j) = \rho^{|i-j|} \text{ for } i,j \in [\![1, m]\!]. \tag{5.23}$$

In the first case, two values have been chosen for ρ, that are $\rho_1 = 0.1$ and $\rho_2 = 0.5$, giving the following centroids:

$$\bar{\mathbf{M}}_1 = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{M}}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}. \tag{5.24}$$

Moreover, the dispersion parameter σ has been fixed to 0.1. The histogram of z has been plotted in Figure 5.6.a showing the independence of z with respect to centroids. A similar experiment has been performed to illustrate the independence of z with respect to σ, shown in Figure 5.6.b. In this case, ρ has been fixed to 0.5 and two dispersion parameters have been considered: $\sigma_1 = 0.1$ and $\sigma_2 = 0.5$.



(a)

(b)

Figure 5.6: Independence of z with respect to (a) $\bar{\mathbf{M}}$ and (b) σ.

It has to be mentioned that the use of z is equivalent to the standardization step $z = \frac{x-\mu}{\sigma}$ for a univariate normal distribution.

Next, by using (5.21), the MAD is given by:

$$\frac{1}{2} = \phi\left(\frac{\text{MAD}}{m\sigma}\right), \tag{5.25}$$

where $\phi(\,\cdot\,)$ is the cumulative distribution function of z. The estimated dispersion parameter can be obtained as:

$$\hat{\sigma} = \frac{1}{\phi^{-1}\left(\frac{1}{2}\right)} \frac{\text{MAD}}{m}, \tag{5.26}$$

where $\phi^{-1}(\,\cdot\,)$ is the inverse cumulative distribution function. In the end:

$$\hat{\sigma} = \frac{K}{m}\text{MAD}, \tag{5.27}$$

with $K = \frac{1}{\phi^{-1}\left(\frac{1}{2}\right)}$. Experiments have shown that $K \approx 1.312$. This value has been obtained on datasets of $N = 10^5$ independent and identically distributed covariance matrices of size $m \times m$, issued from an RGD with ρ fixed to 0.5. Different values have been considered for m: 2, 3, 5, 7 and 16. In addition, for the dispersion parameter σ, 13 values equally sampled between 0.1 and 0.4 have been taken. Next, for each simulated dataset, the MAD is computed according to (5.20) and K is expressed from (5.27). In the end, the mean value of K has been retained, that is approximately 1.312.

By using the proposed algorithm, the estimated threshold T can be obtained without any user intervention. Its value is computed only by taking into consideration the natural variability of the dataset. Note also that this estimated value is an order of magnitude of the threshold we may consider in the Huber estimation algorithm. Depending on the value chosen for constant c in (5.19), different values can be found for the estimated T. All the results presented in this chapter are achieved for $c = 1.5$.

5.4 Performance Analysis

In this section, several tests are performed on simulated data in order to analyze the behavior of the proposed Huber's centroid estimator. The obtained results are presented and they are compared to those given by the center of mass and the median, knowing that the covariance matrices are generated as realizations of RGDs.

Since the centroids are covariance matrices, the manifold of the space of covariance matrices should be taken into account for the estimators' performance evaluation. In the literature, many authors have proposed to define the concept of intrinsic analysis for statistical estimation [Oller & Corcuera 1995, Smith 2005, Garcia & Oller 2006]. To this aim, the concepts of intrinsic root-mean square error (RMSE) and intrinsic bias vector field have been introduced for the Riemannian case. These definitions are recalled next.

Let $\widehat{\mathbf{M}}$ be the estimated centroid of the dataset, that is the estimate of the centroid $\bar{\mathbf{M}}$. The intrinsic RMSE is given by [Oller & Corcuera 1995, Smith 2005, Garcia & Oller 2006]:

$$RMSE = \sqrt{E\left[d^2(\widehat{\mathbf{M}}, \bar{\mathbf{M}})\right]}, \tag{5.28}$$

where $d(\,\cdot\,)$ is the Riemannian distance defined in (5.5). In addition, the bias vector field $b(\bar{\mathbf{M}})$ of $\widehat{\mathbf{M}}$ is given by [Oller & Corcuera 1995, Smith 2005, Garcia & Oller 2006]:

$$\mathbf{b}(\bar{\mathbf{M}}) = \mathrm{Log}_{\bar{\mathbf{M}}} E_{\bar{\mathbf{M}}}\left[\widehat{\mathbf{M}}\right] = E\left[\mathrm{Log}_{\bar{\mathbf{M}}} \widehat{\mathbf{M}}\right], \tag{5.29}$$

knowing that $E_{\bar{\mathbf{M}}}\left[\widehat{\mathbf{M}}\right] = \mathrm{Exp}_{\bar{\mathbf{M}}} E\left[\mathrm{Log}_{\bar{\mathbf{M}}} \widehat{\mathbf{M}}\right]$. Since the bias vector field $\mathbf{b}(\bar{\mathbf{M}})$ in the Riemannian space is a covariance matrix, its norm has to be computed for further evaluation:

$$\|\mathbf{b}(\bar{\mathbf{M}})\| = \mathrm{tr}\left(\left(\bar{\mathbf{M}}^{-1}\mathbf{b}\left(\bar{\mathbf{M}}\right)\right)^2\right), \tag{5.30}$$

where $\mathrm{tr}(\,\cdot\,)$ is the trace operator. For a better understanding of these performance measures, a parallel with the Euclidean space is drawn in Table 5.2.

For all the experiments, the simulated covariance matrix datasets are obtained by using the expression given in (5.23). In addition, $m = 2$ and 5000 Monte Carlo runs are used for the performance evaluation.

The first experiment consists in studying the influence of the dataset's size N on the centroid's estimation performance, for no outlier values. In this case, the dataset contains between 100 and 5000 independent and identically distributed covariance

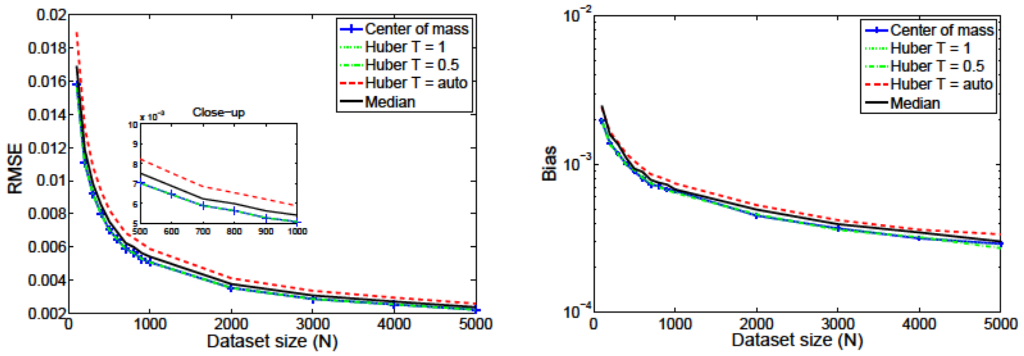
Table 5.2: Definitions of RMSE and bias vector field in both Euclidean and Riemannian spaces.

Euclidean Space	**Riemannian Manifold**
Let $\hat{\theta}$ be the estimate of parameter θ: $RMSE = \sqrt{E[(\hat{\theta}-\theta)^2]}$; $\mathbf{b}(\theta) = E[\hat{\theta}] - \theta$.	Let $\widehat{\mathbf{M}}$ be the estimate of the centroid $\bar{\mathbf{M}}$: $RMSE = \sqrt{E[d^2(\widehat{\mathbf{M}}, \bar{\mathbf{M}})]}$; $\mathbf{b}(\bar{\mathbf{M}}) = \mathrm{Log}_{\bar{\mathbf{M}}} E_{\bar{\mathbf{M}}}[\widehat{\mathbf{M}}]$.

matrices of size 2×2 issued from an RGD having the dispersion $\sigma = 0.1$ and the centroid $\bar{\mathbf{M}}$ obtained from (5.23) for $\rho = 0.7$:

$$\bar{\mathbf{M}} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}. \tag{5.31}$$

Figure 5.7 draws the results obtained for the intrinsic RMSE (Figure 5.7.a) and for the intrinsic bias vector field (Figure 5.7.b), when the centroids are estimated by using the center of mass (in blue), the median (in black) and the Huber's centroid with fixed threshold $T = 1$ and 0.5 (in green) and automatically computed value for T (in red). As expected, as there are no outlier observations in the dataset, the center of mass is slightly better than the other estimators. Moreover, it can be noticed that a higher number N of covariance matrices, gives a better centroid estimation.



(a) (b)

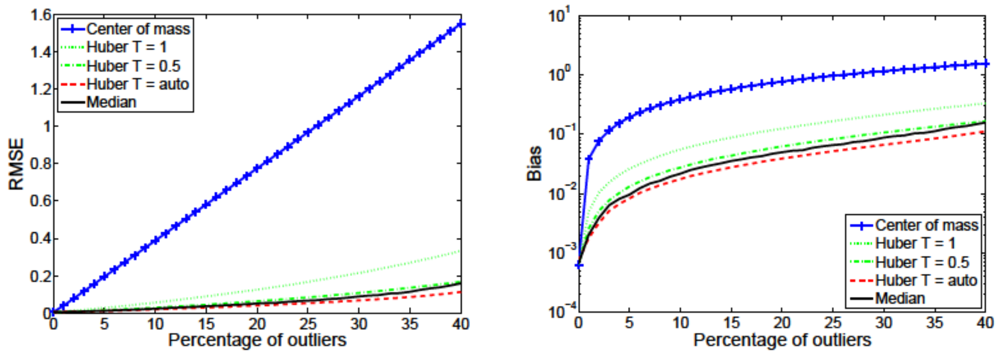
Figure 5.7: (a) The RMSE and (b) the bias vector field as functions of the dataset's size for no outlier data.

The second test studies the influence of outliers on the centroid's estimation. For this purpose, a dataset containing 1000 matrices of size 2×2 is created. These matrices have an RGD distribution of dispersion $\sigma = 0.1$. The centroid $\bar{\mathbf{M}}$ is obtained as in the previous case for $\rho = 0.7$ (5.31). To this original data set, some outliers are added. They are i.i.d. covariance matrices issued from an RGD of centroid $\bar{\mathbf{M}}_{out} = 10 \times M_o$, with M_o obtained from (5.23), for $\rho_o = 0.1$:

$$\bar{\mathbf{M}}_{out} = \begin{bmatrix} 10 & 1 \\ 1 & 10 \end{bmatrix}. \tag{5.32}$$

The dispersion of the outlier sample σ_{out} is set to 0.1.

Figure 5.8 draws the results obtained for the intrinsic RMSE (Figure 5.8.a) and for the intrinsic bias vector field (Figure 5.8.b) as functions of the percentage of outliers. The behavior of the center of mass (in blue), the median (in black) and the Huber's centroid with fixed threshold $T = 1$ and $T = 0.5$ (in green) and automatically computed value for T (in red) are analyzed, when the percentage of aberrant data varies from 0 to 40%. As observed, the center of mass is clearly influenced by the presence of outliers, while for robust estimators, like the median or the Huber's centroid, this influence in less important. In addition, it can be noticed that the Huber's estimator represents a trade-off between the center of mass and the median.



(a) (b)

Figure 5.8: (a) The RMSE and (b) the bias vector field as functions of the outlier percentage (Source: [Ilea *et al.* 2016c] © [2016] IEEE).

5.5 Application to Classification

In this section, the Huber's estimator is used for texture and MEG signal classification. The obtained results are reported and compared to those given by the center of mass and the median.

5.5.1 Application to Texture Image Classification

The first application of the proposed centroid estimator is in the context of texture image classification. The purpose of this experiment is to analyze the influence of aberrant data on the classification accuracy, by using a modified MIT Vision Texture (VisTex) database [Vis].

5.5.1.1 Database

The VisTex database contains 40 texture images considered as 40 different classes. To obtain the database used for the experiment, the same workflow as in the previous Chapter 4 is implemented. First, each image is divided into 169 patches of 128×128 pixels, with an overlap of 32 pixels. Next, outlier samples with altered intensity are introduced in the dataset. For each class, between 0 and 60 patches are introduced by applying a gradient of luminosity. Figure 4.5 shows a texture image from the VisTex database (Figure 4.5.a), one of its patches (Figure 4.5.b) and its corresponding oultier (Figure 4.5.c). In the end, 6760 patches are obtained and used further for the classification.

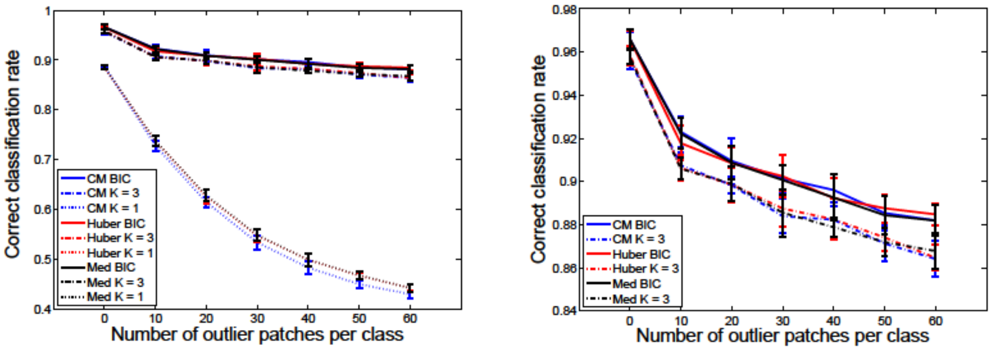
5.5.1.2 Methodology and Results

For this experiment, the classification procedure is based on the spatial dependence of the wavelet coefficients. Thus, each patch is filtered by using the Daubechies' db4 wavelet, with 2 scales and 3 orientations. The spatial dependence is then captured for each pixel of each wavelet subband by considering a vertical and a horizontal spatial neighborhood of 2×1 and 1×2 pixels. Next, the sample covariance matrix (SCM) is estimated for each wavelet subband and both neighborhoods. In the end, a set of $F = 12$ covariance matrices is obtained for every patch.

The database is 100 times equally and randomly divided into a training and a testing set. The elements in the two sets are characterized by F covariance matrices. Further on, each training class c is modeled by a mixture of K RGDs, whose parameters are estimated by using the EM algorithm presented in Chapter 4. Next, a test patch t is affected to the class c maximizing the log-likelohood criterion given in (4.47).

In this experiment, several values are considered for the number of mixture components. First, K is set to 1 meaning that each class contains only one cluster. Therefore, the presence of outliers is not handled by the mixture model and a more accurate analysis of the influence of aberrant data on the centroid estimation methods can be carried out. Second, K is fixed to 3 and third, it is determined by optimizing the BIC criterion given in (4.31).

The classification performances, in terms of overall accuracy, are computed for the center of mass (in blue), the median (in black) and the Huber's centroid with the threshold value T automatically fixed (in red). The results are presented in Figure 5.9 as functions of the number of outlier patches per class, knowing that Figure 5.9.b represents a zoom on the upper part of Figure 5.9.a. By analyzing these graphics, the following conclusions can be drawn. First, as the number of outliers increases, the median and the Huber's estimators perform better than the center of mass. By automatically computing the threshold's value, the Huber's estimator gives classification performances that are close to the median. Second, the results are improved by using the BIC criterion. In the same time, when $K \neq 1$, the dataset's variability can be handled by the mixture model.



(a) (b)

Figure 5.9: (a) Correct classification rate when centroids are estimated by using the center of mass, the median and the Huber's centroid with T automatically fixed and (b) a zoom on the upper part of (a).

5.5.2 Application to MEG Based Brain Decoding

For the first application, outliers have been artificially created and added to the original database. Further on, another example is considered in order to study, this time, the influence of the intrinsic outliers. This second application concerns the brain decoding, based on magnetoencephalography (MEG) data. The database proposed for the Biomag 2014 Decoding Challenge: Brain Decoding Across Subjects (DecMeg2014) [Dec] is used. The idea of brain decoding consists in predicting the stimulus presented to the subject from the concurrent brain activity [Olivetti *et al.* 2014]. For this experiment, two categories of visual stimulus are considered: face and scrambled face. Therefore, the problem to solve can be viewed as a two-class classification task. A detailed description of the neuroscientific experiment implemented to collect the data can be found in [Henson *et al.* 2011].

5.5.2.1 Database

The database contains 16 training and 7 testing subjects. For each training subject, approximately 580 trials are recorded, giving a training set of 9414 trials. Next, for each trial, covariance matrices of size 16×16 are extracted, as described in [Barachant 2014].

5.5.2.2 Methodology and Results

For the classification step, a modified version of the unsupervised method presented in [Barachant 2014] is implemented. For this purpose, a regularized logistic regression model is trained to obtain the initial labels for the unsupervised classification algorithm (k-means). Then, the centroids of each class (face or scrambled face) are computed. At this stage, several estimators are studied: the center of mass, the median, the Huber's estimator with fixed threshold ($T = 0.2$ and $T = 0.5$) and also with automatically computed value for T. In addition, the trimmed based methods [Uehara *et al.* 2016] are considered for $\alpha = 5\%$ of discarded extreme data. Next, for each testing subject, covariance matrices are computed and the classification is performed by two approaches:

- For the first one, the covariance matrices are modeled as mixtures of K RGDs and each test trial $\mathbf{M}_t$ is assigned to the centroid c maximizing the log-likelihood criterion:

$$\arg\max_{c} \left\{ \log \hat{\varpi}_k^c - \log Z(\hat{\sigma}_k^c) - \frac{d^2(\mathbf{M}_t, \bar{\mathbf{M}}_k^c)}{2(\hat{\sigma}_k^c)^2} \right\}, \tag{5.33}$$

  $k = 1, \dots, K$.

  This decision criterion corresponds to the quadratic discriminant analysis mentioned in Chapter 4.

- For the second one, the winner method of the DecMeg2014 competition is considered, for which the test trials are assigned to the closest class, by using the minimum distance to mean (MDM) Riemannian classifier [Barachant *et al.* 2012]:

$$\arg\min_{c} \left\{ d^2(\mathbf{M}_t, \bar{\mathbf{M}}_k^c) \right\}, \tag{5.34}$$

  $k = 1, \dots, K$.

  This approach can be interpreted as the maximization of the log-likelihood (5.33), by considering the homoscedasticity hypothesis and it corresponds to the linear discriminant analysis, mentioned in Chapter 4.

The purpose of the preformed tests is to compare the behavior of the centroid estimators presented in this chapter, but also to study the influence of the dispersion parameter on the classification results.

The obtained results are shown in Table 5.3, where the first column corresponds to the RGD mixture model and the second one to the MDM based method. Further on, several remarks can be made. By analyzing the below table, it can be seen that the use of Huber's estimator provides comparable or even better classification performances than the other robust estimators, but without their disadvantages: division by zero for the median, or choice of the percentage α of discarded observation for the trimmed estimators. Interestingly, note that the estimated Hubers's threshold T is recomputed at each k-means iteration. In this experiment, it varies between 0.38 and 0.46 across the test subjects and the classes. Moreover, the proposed estimated value of T by the MAD gives an order of magnitude of the threshold we may consider in the Huber estimation algorithm. This value can be readjusted to improve the classification performance as observed in Table 5.3.

Table 5.3: Classification results for MEG based brain decoding.

Estimator	RGD (5.33)	MDM (5.34)
CM	73.845	**74.106**
Med	74.150	73.627
Huber $T = 0.2$	**75.109**	**74.847**
Huber $T = 0.5$	73.976	74.063
Huber $T = auto$	74.106	74.455
CM($\text{Trim}_\alpha^{\text{mean}}$) (5.10)	73.888	73.976
CM($\text{Trim}_\alpha^{\text{med}}$) (5.11)	74.237	73.801
Med($\text{Trim}_\alpha^{\text{mean}}$) (5.12)	74.542	74.412
Med($\text{Trim}_\alpha^{\text{med}}$) (5.13)	74.586	74.237

5.6 Influence of Covariance Matrix and Centroid Estimators on Classification

5.6.1 General Remarks

When modeling images or signals by covariance matrices for classification purposes, robust estimators can be used at two different levels:

- during the covariance matrix estimation stage (fixed point estimator, the Huber's estimator, etc.);
- during the centroid estimation stage (median, Huber's centroid, etc.).

In the following, a comparison between these two aspects is performed.

The main difference between these estimators is the fact that they operate at different levels in the classification procedure, illustrated in Figure 5.10. Starting from the initial observation dataset, the features characterizing each observation are extracted as a result of the preprocessing (filtering, wavelet decomposition, etc.) and data modeling (probabilistic models, etc.) steps. These features are covariance matrices and they represent the data's signature. At this stage, robust estimators of covariance matrices are needed, in order to tackle the presence of outlier values in the observations' structure. Next, the estimated covariance matrices are modeled as elements in the Riemannian space and used further in clustering algorithms, like k-means, or Expectation-Maximization. These clustering procedures are based on regrouping the dataset's elements into clusters characterized by their central value. At this point, robust centroid estimators are essential to deal with outliers arising from the inherent variability of the data, or from faulty measurements.

For a better illustration, an example is presented in Figure 5.10, by reconsidering the classification workflow introduced in Chapter 4. In this case, $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ is a set of N independent and identically distributed random vectors according to a parametric model characterized by its covariance matrix. Several observations have to be made on this dataset. First, for each vector $\mathbf{X}_i$, $i = 1, \ldots, N$ the normal observations are represented by yellow circles. Second, the vector $\mathbf{X}_3$ contains some outlier values, displayed as red squares. Therefore, the robust estimators of the covariance matrix are used to reduce their impact in the estimation process. In addition, the vector $\mathbf{X}_j$, marked in green, is itself an aberrant observation arisen, for example, from faulty measurements. In this case, the estimated covariance matrix $\mathbf{M}_j$ will be an outlier in the covariance matrix set. In order to reduce its influence on the centroid's computation, robust estimators of $\bar{\mathbf{M}}$ are needed.

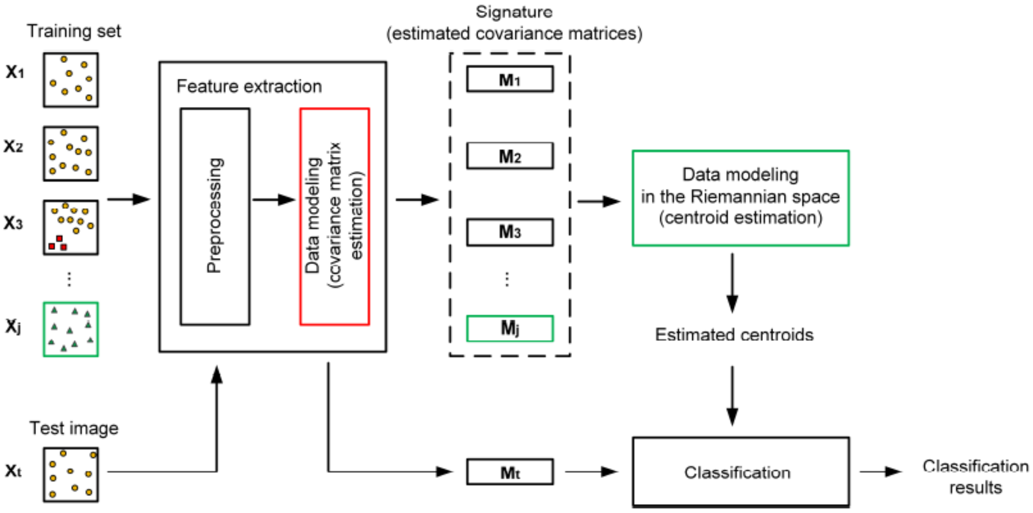

Training set
X1
X2
X3
Xj
Feature extraction
Preprocessing
Data modeling (covariance matrix estimation)
Signature (estimated covariance matrices)
M1
M2
M3
Mj
Data modeling in the Riemannian space (centroid estimation)
Estimated centroids
Test image
Xt
Mt
Classification
Classification results

Figure 5.10: Covariance matrix estimation and centroid estimation steps in the classification workflow.

In order to quantify the impact of outliers, the next subsection introduces an experiment on images simulated for the PolSARproSim software.

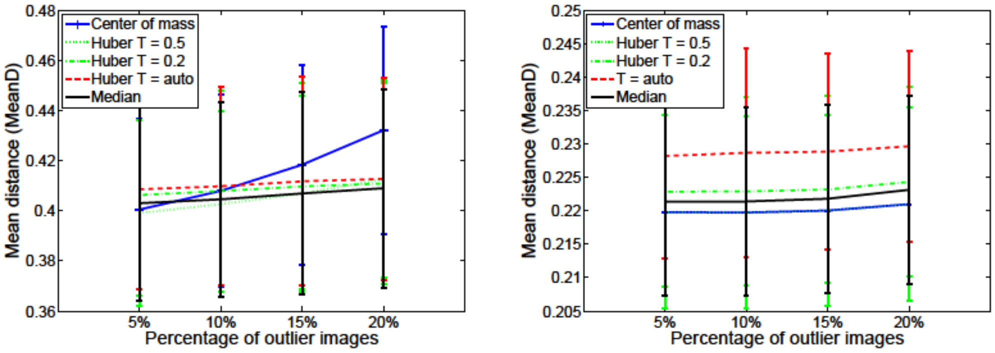
5.6.2 PolSARpro Image Classification

The influence of covariance matrix estimation and centroid estimation is analyzed next, on simulated SAR images. These images are generated by using the PolSARproSim [Williams 2006] software package as described in Chapter 3. In addition, outlier images are created and added to the initial database. The outliers represent forest stands that have been damaged by storms, illnesses, human actions, etc. Therefore, a predefined number of pixels are modified to mimic the consequences of these events. The procedure for obtaining this type of images is detailed in Appendix A.

The first experiment studies the impact of the number of outlier images on the centroid estimation algorithms. First, 40 images of pine forests having less than 10 years old are considered and equally divided into a training and a testing set. Second, images containing aberrant pixels are added only to the training set. The percentage of modified pixels is fixed to 10%, while the number of images containing this type of modification varies from 5% to 20%. Next, the spatial dependence on the wavelet coefficients is modeled by multivariate Gaussian distributions. The covariance matrices are estimated using both SCM and FP estimators. Further on, the central value $\bar{\mathbf{M}}$ of the covariance matrices in the training dataset is computed by using the center of mass, the median and the Huber's centroid with T fixed to 0.2, 0.5 and automatically determined. The geodesic distance between the covariance matrices in the testing set $\mathbf{M}_{t_i}$, $i = 1, \ldots, 20$ and the centroid of the training set is computed. In the end, the mean value of all distances is computed:

$$\mathrm{MeanD} = \frac{1}{20} d(\mathbf{M}_{t_i}, \bar{\mathbf{M}}). \tag{5.35}$$

This algorithm is iterated 100 times, for 100 different partitions of the initial database in training and testing sets. The obtained results are shown in Figure 5.11 for both the SCM (Figure 5.11.a) and the FP (Figure 5.11.b). For the SCM it can be noticed that when the percentage of outlier images is small, all the tested methods give similar distance values. Moreover, for larger values, the robust aspect of the median and the Hubers's centroid can be observed. On the other hand, these remarks are no longer true for the FP estimator. In this case, the mean distance does not vary with the percentage of outliers, or with the centroid estimation method. In conclusion, when the SCM method is used, it is necessary to consider robust centroid estimators in order to obtain results that are independent of the quantity of aberrant data. Note that similar conclusions have also been drawn in the PhD thesis of P. Formont for the classification of textured PolSAR images [Formont 2013].

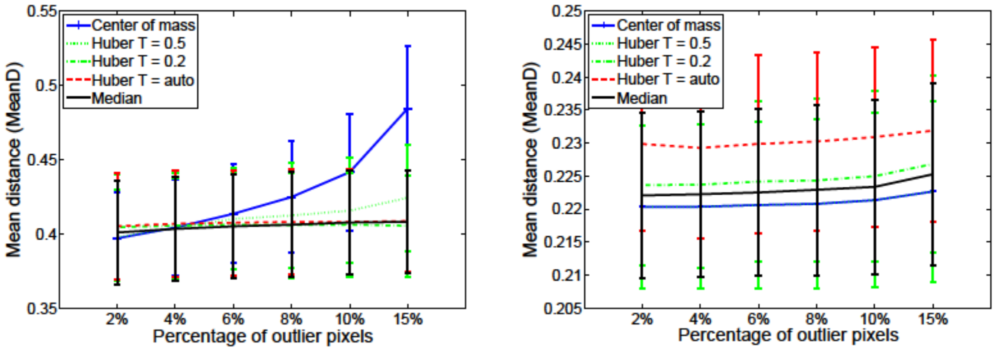


(a) (b)

Figure 5.11: Comparison between (a) the SCM and (b) the FP estimators for a fixed number of aberrant pixels and different numbers of outlier images.

In the second experiment, the influence of the number of aberrant pixels is analyzed. Therefore, the number of outlier images is fixed to 20%. The same workflow as in the first experiment is followed, knowing that the percentage of aberrant pixels varies from 2% to 15%. The results are shown in Figure 5.12 for the SCM (Figure 5.12.a) and the FP (Figure 5.12.b). Conclusions similar to the previous experiment can be drawn: for the SCM estimator, the robust methods for central value estimation become useful for large percentage of aberrant pixels, while no change is observed for the FP estimator.

For these two experiments, the centroid estimation algorithms do not modify the results obtained for the FP estimator. This behavior can be explained by the



(a) (b)

Figure 5.12: Comparison between (a) the SCM and (b) the FP estimators for a fixed number of outlier images and different percentages of aberrant pixels.

fact that the outliers are similar to vector $\mathbf{X}_3$ in Figure 5.10 and therefore, their influence is canceled by using robust covariance matrix estimators during the feature extraction stage. In other words, the FP estimator is able to eliminate the influence of aberrant pixels when image signatures are computed.

The third experiment is designed in order to show the influence of outliers, when the covariance matrices are estimated with the FP algorithm. In this case, outliers similar to vector $\mathbf{X}_j$ in Figure 5.10 are built. The training set contains 20 images of forest stands having less than 10 years old. Among them, 20% have 15% of aberrant pixels. In addition, some outlier images (for example, mislabeled data) of forest stands having between 20 and 30 years old are added. These images do not have aberrant pixels. The same workflow as for the first experiment is followed and the results are shown in Figure 5.13, knowing that the X-axis is not linear. By adding images that are totally different from the majority, the corresponding

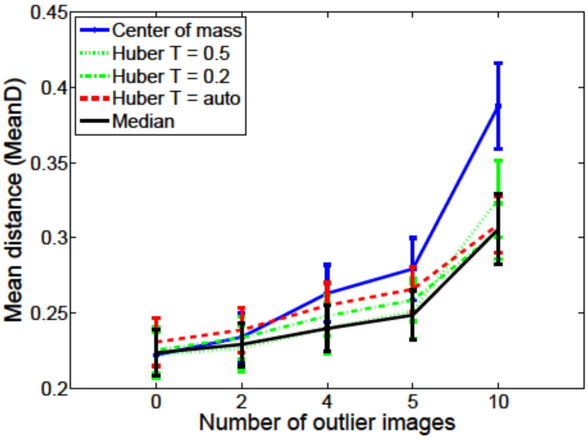


Figure 5.13: Comparison between the centroid estimation methods, when the FP covariance matrices estimator is used.

estimated covariance matrices become outliers for the centroid estimation process. Therefore the importance of the median and Huber's estimator can be also proved for the FP covariance matrix estimator. Even though, for large number of outliers, the estimation performances diverge from the results obtained for outlier-free data, the robust estimators remain less influenced than the center of mass.

In conclusion, depending on the nature of outliers, their influence in the estimation process can be diminished by choosing the appropriate robust approach.

5.7 Conclusions and Perspectives

5.7.1 Conclusions

Many signal and image processing applications, like classification [Said *et al.* 2015a], segmentation [Gu *et al.* 2014], or filtering [Barbaresco *et al.* 2013] require the computation of the central value of a covariance matrix dataset. In this chapter, several ideas concerning the robust centroid estimation on the manifold of covariance matrices have been presented.

First, a new method, called the Huber's centroid, for the estimation of the central value $\bar{\mathbf{M}}$ of a covariance matrix dataset has been introduced, based on the Huber's cost function. This estimator is defined as a trade-off between the center of mass and the median, where the first one is efficient for datasets with no outliers, while the second one is robust to the presence of aberrant observations. The contribution of outliers in the estimation process is controlled by the cost function's parameter, that is the threshold T.

Second, a gradient descent-based algorithm on the manifold of covariance matrices has been proposed for the centroid's computation.

Third, a method to automatically compute the Huber's threshold T has been developed. This method is based on the concept of median absolute deviation that has been generalized to the Riemannian case. Thus, experiment-dependent thresholds can be obtained, in order to capture the intrinsic variability presented in each dataset. By using this approach, an order of magnitude of threshold T is found, which may give a clue on the value that has to be considered in the Huber's estimation algorithm.

Further on, the properties of the Huber's centroid, have been analyzed on simulated data. The robustness to outliers and the influence of the dataset's size in the estimation process have been investigated. A comparison with the center of mass and the median has been also performed.

Next, the Huber's centroid has been used in the context of two real data classification problems, that are the texture image classification and the brain decoding. The results have been compared to some state-of-the-art methods that are the center of mass, the median and the trimmed based estimators.

In the end, a parallel is made between the two types of robust estimators that may be involved in the classification workflow: the covariance matrix estimators and centroid estimators.

5.7.2 Perspectives

Further work will include several directions:

- *The computation algorithm for the Huber's centroid:* in Section 5.3.2 a gradient descent-based method has been proposed to estimate the Huber's centroid. Future works will analyze the convergence of this algorithm and the existence of a unique solution. It has to be mentioned that for the performed experiments, no convergence problems have been encountered. In addition, the convergence of the gradient descent algorithm has been already proved for the center of mass [Karcher 1977] and the median [Yang 2010]. However, a mathematical proof will be searched for the Huber's centroid.

- *The concept of MAD for Riemannian manifolds*: in Section 5.3.3, the MAD has been introduced and a link between it and the dispersion parameter σ has been defined. For this purpose, the transformation $z = \frac{d(\mathbf{M}_i, \bar{\mathbf{M}})}{m\sigma}$ has been proposed and its statistics have been studied. Experimentally, it has been observed that z seems to have a chi-squared distribution and that it is independent of the dispersion σ and the centroid $\bar{\mathbf{M}}$. Further on, a mathematical proof for these observations will be searched, in order to find an explicit value for the coefficient K linking the MAD and σ. It has to be reminded that previously, K has been determined from experiments and set to $K \approx \mathbf{1.312}$.

- *The generalization to other types of centroid estimators*: in the present chapter, the Huber's function has been introduced for centroid estimation. Nevertheless, the proposed method may be generalized to other functions in order to develop new robust centroid estimators. For instance, the family of Hampel functions [Hampel *et al.* 2005], the linear quadratic quadratic (LQQ) function [Koller & Stahel 2011], the Welsh function [Maronna *et al.* 2006] may be considered for future work.

Chapter 6

Riemannian Fisher Vectors

Contents

6.1 Introduction

In the previous chapters, we have presented different robust classification algorithm based on global features. Even if these global descriptors have provided relatively good performances, these features are not adapted to non-stationary signals (local deformation, ...). To face this issue, many researchers turned their attention to local descriptors. Bag of words (BoW), Fisher vectors (FV), or vectors of locally aggregated descriptors (VLAD) are examples of local models used to capture the information lying in signals [Jaakkola & Haussler 1998], images [Sánchez *et al.* 2013], or videos [Faraki *et al.* 2015b]. These descriptors have multiple advantages. First, the obtained information can be used in a wide variety of applications, like classification [Sánchez *et al.* 2013] and categorization [Perronnin & Dance 2007], text [Salton & Buckley 1988] and image [Douze *et al.* 2011] retrieval, action and face recognition [Faraki *et al.* 2015a], etc. Second, combined with powerful local feature descriptors, such as SIFT, they are robust to transformations like scaling, translation, or occlusion [Faraki *et al.* 2015a].

These three approaches, have been widely used for many applications involving non-parametric features. Recently BoW and VLAD have been extended to the case where each feature is a point on a Riemannian manifold. This is, for instance, the case where local descriptors are covariance matrices. This includes many different applications in image processing, like classification [Barachant *et al.* 2013, Said *et al.* 2015a, Ilea *et al.* 2015b], image segmentation [Garcia & Oller 2006], object detection [Mader & Reese 2012, Robinson 2005], etc. In [Faraki *et al.* 2015b] and [Faraki *et al.* 2014], the BoW approach has been extended to the so-called log-Euclidean bag of words (LE-BoW) and bag of Riemannian words (BoRW) models by considering the log-Euclidean and the geodesic distance between two points on the manifold. In addition, the Riemannian version of the VLAD model (R-VLAD) has been developed in [Faraki *et al.* 2015a] and has shown superior classification performances, compared to the classic VLAD.

Until now, FV have not been yet generalized in the same manner to Riemannian manifold, due to the lack of probabilistic generative models suited for parametric descriptors. In Chapter 4, it has been shown that the covariance matrices are elements on the manifold that can be modeled by Riemannian Gaussian distributions [Said *et al.* 2015b] and Riemannian Laplace distributions [Hajri *et al.* 2016]. The present chapter proposes an application of these distributions to model local descriptors by introducing the Riemannian Fisher Vectors (RFV). In this context, the theoretical background is fixed and it is validated on classification problems.

The chapter is structured as follows. Section 6.2 presents the general classification workflow, when local features are used for information modeling. An overview of the BoW, the FV and the VLAD descriptors defined on the Euclidean space is also given. In addition, the methods that extend the BoW and VLAD to the Riemannian manifold are also described. Section 6.3 introduces the proposed extension of FV to the Riemannian manifold, resulting in the Riemannian Fisher vectors. These descriptors are defined for both the Gaussian and Laplace mixture models and their relation with the R-VLAD is detailed. Section 6.4 presents an application of the proposed method to texture image classification, based on region covariance descriptors. For this experiment, in order to speed-up the computation of covariance matrices, they are estimated by using the integral images. Moreover, the influence of the classification method on the RFV is analyzed, by comparing the support vector machine and random forest classifiers. In the end, Section 6.5 reports some conclusions and perspectives.

6.2 Local Features for Information Modeling

The work presented in this chapter focuses on classification based on local features. In this context, the information modeling process and the classification workflow are illustrated in Figure 6.1, where four different stages can be identified: feature extraction, codebook creation, coding and post-processing, and classification.

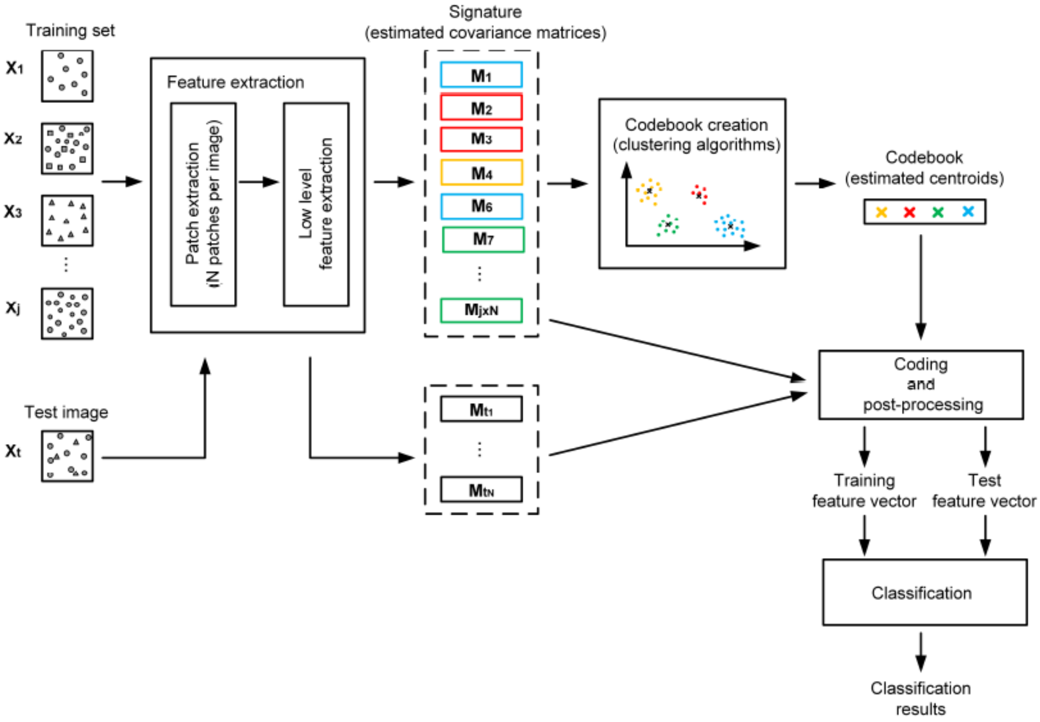

Training set
X_1
X_2
X_3
X_j
Feature extraction
Patch extraction (N patches per image)
Low level feature extraction
Signature (estimated covariance matrices)
M_1
M_2
M_3
M_4
M_6
M_7
M_{jxN}
Codebook creation (clustering algorithms)
Codebook (estimated centroids)
Test image
X_t
M_{t1}
M_{tN}
Coding and post-processing
Training feature vector
Test feature vector
Classification
Classification results

Figure 6.1: Classification workflow for local features based methods.

During the first stage, some characteristics, called low level features, are extracted, from each element in the database. These descriptors are often computed on patches and as a result, a set of feature vectors, or signatures, is obtained for each element in the database. Further on, supervised classification algorithms will be used, and therefore, this initial set of feature vectors is divided into two sets, called training and testing sets.

The codebook creation stage is performed on the training set and its purpose is to identify the significant features from the dataset. Usually, this procedure is performed by means of clustering algorithms, like k-means, or expectation-maximization. By using these algorithms, the set is partitioned into a predefined number of clusters, each of them being described by parameters, such as the cluster's centroid, the dispersion, the associated weight, etc. The obtained features are called codewords and they are grouped in a codebook, also called a dictionary.

Based on the extracted codebook, the coding stage is implemented next, by projecting the training signature set onto the codebook space. The purpose of this operation is to express the signatures by using the previously obtained codewords. As stated earlier, in the introduction, approaches like bag of words, Fisher vectors, or vectors of locally aggregated descriptors can be used, resulting in some local models that capture the underlying information. After their computation, a post-processing step is often applied, consisting in two possible normalizations, namely the ℓ_2 [Perronnin *et al.* 2010b] and power normalizations [Perronnin *et al.* 2010a]. The coding process will be detailed in the following sections, for each of these methods.

For the final classification stage, the testing feature set is also mapped onto the codebook space. The classification results are obtained, in the end, by associating the test images to the class of the most similar training observation, according to some decision rules. In practice, algorithms like k-nearest neighbors, support vector machine, random forest, etc. are used.

In the following, a short description of the coding models (i.e. bag of words, Fisher vectors, and vectors of locally aggregated descriptors) for features living in the Euclidean space is given. In addition, some models that extend the BoW [Faraki *et al.* 2015b, Faraki *et al.* 2014] and VLAD [Faraki *et al.* 2015a] to the Riemannian manifolds are also presented.

6.2.1 Euclidean Space

6.2.1.1 Bag of Words

The *bag of words* (BoW) model has been used for text retrieval and categorization [Salton & Buckley 1988, Joachims 1998] and then extended to visual categorization [Csurka *et al.* 2004]. In the context of text analysis, the BoW approach has been used to model a text by a histogram containing the number of occurrences of each word. This idea has been applied to image characterization, where the "words" are represented by some discriminating features. Therefore, the image is described by the number of occurrences of these patterns.

The BoW model follows the general workflow presented earlier, in the introductory part. First, the codebook is created during the learning stage. Next, based on the extracted codewords, the data space is partitioned in Voronoï regions, by assigning each data point to the closest centroid. Further on, for each element in the dataset, its signature is determined by computing the histogram of the number of occurrences of each codeword in its structure, as shown in Figure 6.2. In the end, the classification is performed by means of a distance measure between two

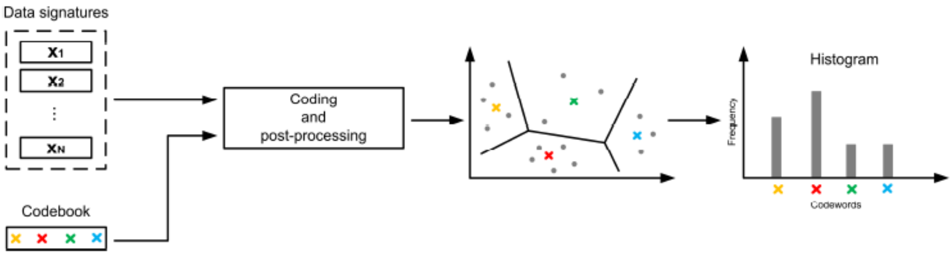


Figure 6.2: Feature vector computation for the bag of words method.

histograms, like the chi-squared distance.

The BoW method has several advantages. This is a simple and computational effective method, invariant to affine transformations and occlusions [Csurka *et al.* 2004]. On the other hand, its performances depend on the codebook's size, the best results being obtained for large dictionaries [Perronnin & Dance 2007]. In addition, the BoW method counts only the number of local descriptors assigned to each Voronoï region. Thus, the classification results may be improved by including other statistics, such as the variance of local descriptors. This is the case of Fisher vectors that are presented next.

6.2.1.2 Fisher Vectors

Fisher vectors (FV) are descriptors based on Fisher kernels [Jaakkola & Haussler 1998], representing methods for measuring if samples are correctly fitted by some given models. By using FV, a sample is characterized by the gradient vector of the probability density function that models it. Classically, a Gaussian mixture model (GMM) [Perronnin & Dance 2007] is considered. In practice, the probability density function is replaced by the log-likelihood and, as mentioned in [Perronnin & Dance 2007], its gradient describes the direction in which parameters should be modified to best fit the data. In other words, the gradient of the log-likelihood with respect to a parameter describes the contribution of that parameter to the generation of a particular observation [Jaakkola & Haussler 1998]. A large value of this derivative is equivalent to a large deviation from the model. Further on, that can be translated into the fact that the model does not correctly fit the data.

Let $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_n\}_{n=1:N}$, with $\mathbf{x}_n \in \mathbb{R}^m$, be a sample of N low level m-dimensional features extracted from a dataset. These features are modeled as i.i.d realizations of the parametric model $p(\boldsymbol{\mathcal{X}}|\theta)$. By extracting the FV for this set, the sample $\boldsymbol{\mathcal{X}}$ is projected onto a fixed length vector, whose size depends on the number of parameters in θ. More precisely, through the Fisher kernels, the sample is characterized by its deviation from the model [Sánchez *et al.* 2013]. This deviation is measured by computing the Fisher score $U_{\boldsymbol{\mathcal{X}}}$ [Jaakkola & Haussler 1998], that is the gradient ∇ of the log-likelihood with respect to the model's parameters θ:

$$U_{\boldsymbol{\mathcal{X}}} = \nabla_\theta \log p(\boldsymbol{\mathcal{X}}|\theta) = \nabla_\theta \sum_{n=1}^{N} \log p(\mathbf{x}_n|\theta). \tag{6.1}$$

In classification problems, the gradient of the log-likelihood can be normalized by using the Fisher information matrix F_θ [Jaakkola & Haussler 1998]. For this purpose, F_θ is given by:

$$F_\theta = E_{\mathcal{X}}[U_{\mathcal{X}} U_{\mathcal{X}}^T], \tag{6.2}$$

where $E_{\mathcal{X}}[\,\cdot\,]$ denotes the expectation over $p(\mathcal{X}|\theta)$ and $(\,\cdot\,)^T$ is the transpose operator. Therefore, the normalized Fisher score becomes [Perronnin & Dance 2007]:

$$F_\theta^{-1/2} \nabla_\theta \log p(\mathcal{X}|\theta). \tag{6.3}$$

Often, this normalization step is not performed in practice, due to the associated computational costs. In this case, F_θ is approximated by the identity matrix. Nevertheless, in [Perronnin & Dance 2007], the authors have shown that by performing the normalization, the performances are increased.

Let $\mathcal{X} = \{\mathbf{x}_n\}_{n=1:N}$ be an N-sample of m-dimensional observations modeled as a Gaussian mixture model with K components. Thus:

$$p(\mathbf{x}_n|\theta) = \sum_{k=1}^{K} \varpi_k p(\mathbf{x}_n|\mu_k, \mathbf{M}_k), \tag{6.4}$$

where $\theta = \{(\varpi_k, \mu_k, \mathbf{M}_k)_{1\leq k \leq K}\}$ is the parameter vector for the k^{th} component. ϖ_k is the mixture weight, with $\varpi_k \in (0,1)$ and $\sum_{k=1}^{K} \varpi_k = 1$, μ_k is the mean vector, $\mathbf{M}_k$ is the covariance matrix and

$$p(\mathbf{x}_n|\theta_k) = \frac{1}{(2\pi)^{\frac{m}{2}} |\mathbf{M}_k|^{\frac{1}{2}}} \exp\Big\{ -\frac{1}{2}(\mathbf{x}_n - \mu_k)'\mathbf{M}_k^{-1}(\mathbf{x}_n - \mu_k)\Big\}. \tag{6.5}$$

In addition, the covariance matrix is assumed to be diagonal and $\sigma_k^2 = \text{Diag}(\mathbf{M}_k)$ is the variance vector.

Next, the derivatives of each dimension d, $d = 1, \ldots, m$ with respect to θ are computed, by taking into consideration two observations:

- the probability $\gamma_k(\mathbf{x}_n)$ that the observation $\mathbf{x}_n$ is generated by the k^{th} Gaussian component is computed as:

$$\gamma_k(\mathbf{x}_n) = \frac{\varpi_k \, p(\mathbf{x}_n|\theta_k)}{\sum_{j=1}^{K} \varpi_j \, p(\mathbf{x}_n|\theta_j)}; \tag{6.6}$$

- to ensure the constraints made on the mixture weights, the following parametrization is generally adopted:

$$\varpi_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{K} \exp(\alpha_j)}. \tag{6.7}$$

As a result, by neglecting the Fisher information matrix, the gradients of the log-likelihood are obtained as:

$$\frac{\partial \log p(\boldsymbol{\mathcal{X}}|\theta)}{\partial \mu_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n) \left(\frac{\mathbf{x}_n^d - \mu_k^d}{\left(\sigma_k^d\right)^2} \right), \tag{6.8}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{X}}|\theta)}{\partial \sigma_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n) \left[\frac{\left(\mathbf{x}_n^d - \left(\mu_k^d\right)^2\right)}{\left(\sigma_k^d\right)^3} - \frac{1}{\sigma_k^d} \right], \tag{6.9}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{X}}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{x}_n) - \varpi_k\right]. \tag{6.10}$$

By using some, or all of these derivatives, the Fisher vectors are obtained, as illustrated in Figure 6.3. In [Sánchez *et al.* 2013], it has been shown that the combination between the derivatives with respect to the mean and dispersion gives the most discriminating descriptors.

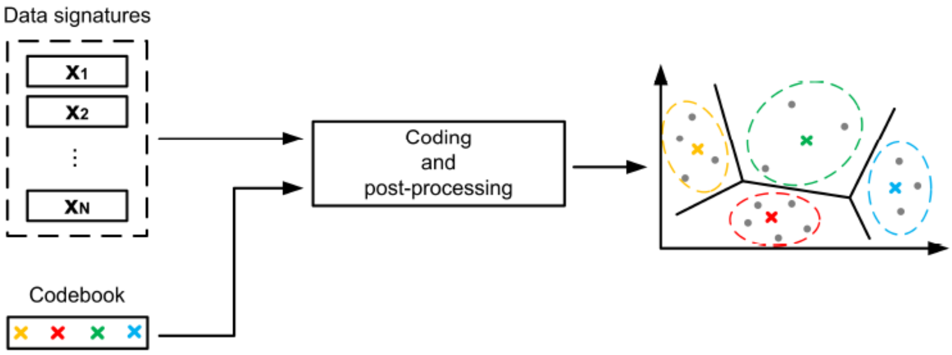


Figure 6.3: Feature vector computation for the Fisher vector method.

6.2.1.3 Vectors of Locally Aggregated Descriptors

The *vectors of locally aggregated descriptors* (VLAD) represent a simplification of the Fisher kernel [Jégou *et al.* 2010], based on the definition of the codebook.

Let $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_n\}_{n=1:N}$, with $\mathbf{x}_n \in \mathbb{R}^m$, be an N-sample of low level m-dimensional features extracted from a dataset. This set is partitioned into K clusters, given by their centroids usually determined by the k-means algorithm. For each cluster c_k, $k = 1, \dots, K$, a vector containing the differences between the cluster's centroid μ_k and each element $\mathbf{x}_n$ in that cluster is computed. Next, the sum of differences concerning each cluster c_k is determined:

$$\mathbf{v}_k = \sum_{\mathbf{x}_n \in c_k} \mu_k - \mathbf{x}_n, \tag{6.11}$$

as shown in Figure 6.4. In the end, the final VLAD descriptor is given by the concatenation of all the previously obtained sums:

$$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_K], \tag{6.12}$$

which leads to very good results in practice [Jégou *et al.* 2010].

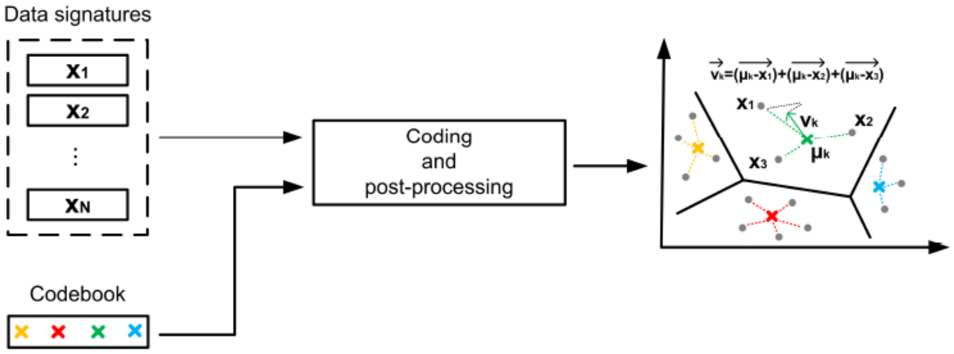

Data signatures
x1
x2
⋮
xN
Codebook
Coding and post-processing
$\vec{v_k}=(\overrightarrow{\mu_k-x_1})+(\overrightarrow{\mu_k-x_2})+(\overrightarrow{\mu_k-x_3})$
x1
vk
x2
x3
μk

Figure 6.4: Feature vector computation for the VLAD method.

The VLAD descriptors can be also obtained starting from FV, by taking into consideration only the derivatives with respect to the GMM mean, given in (6.8). In addition, the homoscedasticity assumption ($\sigma_k = \sigma$, $\forall k = 1, \dots, K$) and the hard assignment scheme ($\gamma_k(\mathbf{x}_n) = 1$ if $\mathbf{x}_n \in c_k$ and 0 otherwise) are required to obtain the VLAD [Sánchez *et al.* 2013, Jégou *et al.* 2010].

Recently, the BoW and VLAD methods have been generalized to the Riemannian case. In the following, these extensions to the manifold of covariance matrices are presented.

6.2.2 Extension to Riemannian Manifolds

6.2.2.1 Bag of Words on the Riemannian Manifold

In [Faraki *et al.* 2014] and [Faraki *et al.* 2015b], the BoW approach has been extended to the so-called bag of Riemannian words (BoRW) and log-Euclidean bag of words (LE-BoW) models.

These descriptors have been obtained by addressing two problems:

- the codebook construction in the Riemannian space;
- the histogram construction in the Riemannian space.

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be a sample of N i.i.d observations on the Riemannian manifold.

In order to take into account the geometry of the covariance matrices space, in [Faraki *et al.* 2014] the authors have extended the k-means algorithm to the Riemannian space, by using the Rao's geodesic distance between two points on the manifold. In this case, the cluster's centroid $\bar{\mathbf{M}}$ is given by the center of mass obtained by minimizing the cost function in (5.4), recalled next:

$$f_{CM}(\bar{\mathbf{M}}) = \frac{1}{N}\sum_{n=1}^{N} d^2(\bar{\mathbf{M}}, \mathbf{M}_n), \tag{6.13}$$

where $d(\cdot)$ represents the Rao's Riemannian distance [James 1973]. Next, the histogram has been computed by associating each local descriptor to its closest codeword in terms of geodesic distance.

A slightly different algorithm can be obtained starting from the one given in [Faraki *et al.* 2014], by introducing the RGD, or RLD, probability density function in the codebook construction.

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be an N-sample of i.i.d observations issued from one of the Riemannian distributions. In this case, the data space is partitioned in K Voronoï regions by maximizing the corresponding probability density function. More precisely, each observation $\mathbf{M}_n$ is assigned to the cluster k, $k = 1, \ldots, K$ according to:

$$\arg\max_{k} p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k), \tag{6.14}$$

where $p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)$ is the RGD, or RLD probability density function given in (4.8), or (4.34).

In [Faraki *et al.* 2015b] the authors have proposed another approach, called the LE-BoW, which implies the transformation of the matrix space into a vector space, by means of log-Euclidean representations. As a result, the codebook can be obtained by the classical k-means defined in the Euclidean space and the histogram is then built in the log-Euclidean space.

6.2.2.2 Riemannian Vectors of Locally Aggregated Descriptors

The Riemannian version of VLAD, called Riemannian Vectors of Locally Aggregated Descriptors (R-VLAD), has been developed in [Faraki *et al.* 2015a] and has shown superior classification performances, compared to the classic VLAD algorithm.

In order to define this descriptor, two problems had to be addressed first:

- the definition of a metric for the clustering algorithm;
- the definition of the Riemannian subtraction.

In [Faraki *et al.* 2015a] these issues have been solved by choosing the geodesic distance [James 1973] as a similarity measure and the Riemannian logarithm mapping [Higham 2008] to perform the subtraction on the manifold.

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be an N-sample of i.i.d observations on the Riemannian manifold. Based on the geodesic distance, $\boldsymbol{\mathcal{M}}$ is partitioned in K clusters with the centroids denoted by c_k, $k = 1, \ldots, K$. This partition can be achieved by using the k-means detailed in [Faraki *et al.* 2015a]. In this case, the vector of differences between each centroid c_k and the elements $\mathbf{M}_i \in c_k$, defined in (6.11), becomes:

$$\mathbf{v}_k = \sum_{\mathbf{M}_i \in c_k} \mathrm{Log}_{c_k} \mathbf{M}_i, \tag{6.15}$$

where $\mathrm{Log}(\cdot)$ is the Riemannian logarithm mapping [Higham 2008].

6.3 Riemannian Fisher Vectors

The previous section has described the generalization of the BoW and VLAD models to the Riemannian manifolds. This extension has not yet been done for Fisher vectors, due to the lack of probabilistic generative models, suited for parametric descriptors. Recently, the Riemannian space has been modeled by several distributions and therefore, the Fisher vectors can be defined for the manifold of covariance matrices.

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be an N-sample of i.i.d observations modeled as a mixture of K Riemannian distributions. Under the independence assumption, the probability density function of $\boldsymbol{\mathcal{M}}$ is given by:

$$p(\boldsymbol{\mathcal{M}}|\theta) = \prod_{n=1}^{N} p(\mathbf{M}_n|\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} \varpi_k p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k), \tag{6.16}$$

where $p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)$ represents some density defined on the manifold and $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1\leq k\leq K}\}$ is the parameter vector containing the mixture weight ϖ_k, the central value $\bar{\mathbf{M}}_k$ and the dispersion parameter σ_k.

In order to obtain the Riemannian Fisher Vectors (RFV), the gradient of the probability density function characterizing the data has to be determined. Similar to the Euclidean case, this is achieved by computing the gradient of the log-likelihood with respect to the model parameters. Concerning the gradient's normalization, up to our knowledge, there is no closed-form expression for this Fisher information matrix in the Riemannian space. In practice, it can be estimated by carrying out a Monte Carlo integration. Nonetheless, due to the computation cost of this approach, the Fisher information matrix is often approximated by the identity matrix [Perronnin & Dance 2007].

In the following, RFV are derived for the Riemannian Gaussian model and Riemannian Laplace model. Closed-form expressions of the derivatives of the log-likelihood functions with respect to $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1\leq k\leq K}\}$ can be computed based on the following observations:

- the probability $\gamma_k(\mathbf{M}_n)$ that the observation $\mathbf{M}_n$ is generated by the k^{th} mixture component is computed as:

$$\gamma_k(\mathbf{M}_n) = \frac{\varpi_k \, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \varpi_j \, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}; \tag{6.17}$$

- to ensure the constraints of positivity and sum-to-one for the weights, the derivative of the log-likelihood with respect to this parameter needs the following parametrization [Sánchez *et al.* 2013]:

$$\varpi_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{K} \exp(\alpha_j)}. \tag{6.18}$$

In the end, the vectorized representation of the derivatives of the log-likelihood with respect to the parameters in θ, gives the Riemannian Fisher vectors.

By using the RFV, a sample is characterized by a feature vector containing some, or all the derivatives, having the maximum length given by the number of parameters in θ.

6.3.1 Riemannian Gaussian Model

In order to obtain the RFV for the Riemannian Gaussian model, the probability density function $p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)$ in (6.16) represents the Riemannian Gaussian distribution, introduced in Section 4.3.1 and recalled next:

$$p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) = \frac{1}{Z(\sigma_k)} \exp\Big\{ - \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\}, \tag{6.19}$$

where $\bar{\mathbf{M}}_k \in \mathcal{P}_m$ and $\sigma_k > 0$ are the location and the dispersion parameters. $Z(\sigma_k)$ is a normalization factor independent of the centroid $\bar{\mathbf{M}}_k$ and $d(\,\cdot\,)$ is the Riemannian distance [James 1973].

As a result, the derivatives with respect to the elements in the parameter vector are [Ilea *et al.* 2016b]:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \, \frac{\mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n)}{\sigma_k^2}, \tag{6.20}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left\{ \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} - \frac{Z'(\sigma_k)}{Z(\sigma_k)} \right\}, \tag{6.21}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{M}_n) - \varpi_k \right], \tag{6.22}$$

where $\mathrm{Log}_{\bar{\mathbf{M}}_k}(\,\cdot\,)$ is the Riemannian logarithm mapping [Higham 2008], $\gamma_k(\,\cdot\,)$ and α_k are defined in (6.17), respectively (6.18). $Z'(\sigma_k)$ is the derivative of $Z(\sigma_k)$ with respect to σ_k. All the computational details concerning the derivatives with respect to θ are given in Appendix B. In addition, the method for computing $Z'(\sigma_k)$ has been already presented in Section 4.3.3.

By analyzing these expressions, it can be noticed that they are similar to the ones obtained for the GMM presented in Section 6.2.1.2, more precisely, with the expressions in (6.8), (6.9) and (6.10).

6.3.2 Riemannian Laplace Model

Starting from their initial definition, the Riemannian Fisher vectors are extended next to the Riemannian Laplace distribution. In this case, the probability density function $p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)$ in (6.16) is given by the RLD introduced in Section 4.4.1 and recalled further:

$$p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) = \frac{1}{\zeta(\sigma_k)} \exp\Big\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\}, \tag{6.23}$$

where $\bar{\mathbf{M}}_k \in \mathcal{P}_m$ and $\sigma_k > 0$ are the location and the dispersion parameters. $\zeta(\sigma_k)$ is a normalizing factor independent of $\bar{\mathbf{M}}_k$ and $d(\,\cdot\,)$ is the Riemannian distance [James 1973].

Therefore, the derivatives with respect to the distribution's parameters are [Ilea *et al.* 2016a]:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \gamma_i(\mathbf{M}_n)\, \frac{\operatorname{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n)}{2\,\sigma_k^2\, d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}, \tag{6.24}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left\{ \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} - \frac{\zeta'(\sigma_k)}{\zeta(\sigma_k)} \right\}, \tag{6.25}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{M}_n) - \varpi_k \right], \tag{6.26}$$

where $\operatorname{Log}_{\bar{\mathbf{M}}_k}(\,\cdot\,)$ is the Riemannian logarithm mapping [Higham 2008], $\gamma_k(\,\cdot\,)$ and α_k are defined in (6.17) and (6.18). $\zeta'(\sigma_k)$ is the derivative of $\zeta(\sigma_k)$ with respect to σ_k and its computation has been detailed in Section 4.4.3. The mathematical proof of the expressions in (6.24), (6.25) and (6.26) can be found in Appendix C.

6.3.3 Relation with R-VLAD

As mentioned in the introductory part, VLAD features are a special case of FV. Therefore, R-VLAD can be viewed as a particular case of the proposed RFV. More precisely, R-VLAD is obtained by taking into consideration only the derivatives with respect to the central value $\bar{\mathbf{M}}_k$ given in (6.20), or (6.24). In addition, a hard assignment scheme is applied, knowing that the intrinsic k-means algorithm is usually used for the codebook generation. Starting from the definition of the elements v_k in the R-VLAD descriptor [Faraki *et al.* 2015a] recalled here

$$v_k = \sum_{\mathbf{M}_n \in c_k} \operatorname{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n), \tag{6.27}$$

where $\mathbf{M}_n \in c_k$ are the elements assigned to the cluster c_k, $k = 1, \ldots, K$, the hard assignment implies that:

$$\gamma_k(\mathbf{M}_n) = \begin{cases} 1, & \text{if } \mathbf{M}_n \in c_k \\ 0, & \text{otherwise.} \end{cases} \tag{6.28}$$

Moreover, the assumption of homoscedasticity is considered, that is $\sigma_k = \sigma\ , \forall k = 1, \ldots, K$. By taking into account these two hypotheses, it is clear that (6.20) and (6.24) reduce to (6.27), hence confirming that RFV are a generalization of R-VLAD descriptors. The only difference between (6.20) and (6.24) relies on the way the codebooks are constructed. The former considers that the centroid of each cluster is the center of mass, while the latter assumes that the centroid is the median.

6.4 Application to Texture Image Classification

This section introduces an application to texture image classification. The aim of this experiment is threefold.

The first objective is to analyze the potential of the proposed RFV, for both the Gaussian and Laplace models, compared to the recently proposed bag of Riemannian words (BoRW) model [Faraki *et al.* 2014] and R-VLAD [Faraki *et al.* 2015a]. The BoRW, RFV and R-VLAD are built based on region covariance descriptors [Tuzel *et al.* 2006] containing basic information, like image intensity and gradients. The experiment's purpose is not to find the best classification rates, but to compare the methods, starting from classical descriptors.

The second objective is to determine the RFV that are the most discriminant to retrieve the classes: the one associated to the centroid $\bar{\mathbf{M}}_k$, to the dispersion σ_k or to the mixture weight α_k.

The last objective is to compare two different classification algorithms for the RFV, that are the support vector machine (SVM) and random forest.

6.4.1 Databases

For this work, two texture databases are used:

- *VisTex* [Vis] database illustrated in Figure 2.1, for which each class is composed of 64 images of size 64×64 pixels;
- *Outex_ TC000_ 13* [Out] database shown in Figure 2.2, for which each class is represented by a set of 20 images of size 128×128 pixels.

For both databases, the general classification workflow presented in Section 6.2 is applied and it is detailed next.

6.4.2 Classification Workflow

Earlier in this chapter, it has been shown in Figure 6.1 that the experimental workflow consists in four stages. At the beginning, the descriptors modeling the textural information are extracted. Next, the codebook is generated and the RFV are computed. In the end, a supervised classification algorithm is used to classify these RFV. In the next subsections, each of these stages will be presented.

6.4.2.1 Feature Extraction

For this experiment, the textural information is captured by using region covariance descriptors (RCovDs), obtained from classical features. Thus, for an image I of size $W \times H$, characteristics like the image intensity and the norms of the first and second order derivatives are computed for each pixel $(x, y) \in I$. As a result, a vector $\mathbf{D}$ of $m = 5$ elements is extracted for every pixel [Tuzel *et al.* 2006]:

$$\mathbf{D}(x,y) = \left[I(x,y), \left|\frac{\partial I(x,y)}{\partial x}\right|, \left|\frac{\partial I(x,y)}{\partial y}\right|, \left|\frac{\partial^2 I(x,y)}{\partial x^2}\right|, \left|\frac{\partial^2 I(x,y)}{\partial y^2}\right| \right]^T, \tag{6.29}$$

where $I(x, y)$ is the image intensity of pixel $(x, y) \in I$.

Next, the feature image I_F of image I is built. I_F is a $W \times H \times m$ dimensional array, where each element $I_F(x, y)$ is the m-dimensional vector $\mathbf{D}(x, y)$, as shown in Figure 6.5.

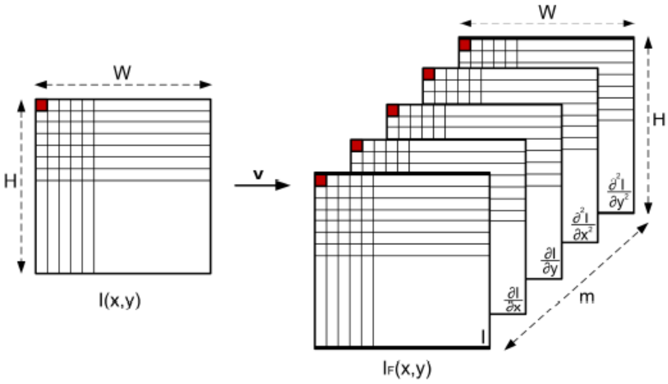


Figure 6.5: Computation of feature images.

Starting from the feature images, the RCovDs are defined as being the estimated covariance matrices $\mathbf{M}_P$ computed on a sliding patch (or region) $P \in I_F$:

$$\mathbf{M}_P = \frac{1}{N_P} \sum_{n=1}^{N_P} (\mathbf{p}_n - \mu)(\mathbf{p}_n - \mu)^T, \tag{6.30}$$

where N_P represents the number of m-dimensional points $\{\mathbf{p}_n\}_{n=1,\dots,N_p}$ in the patch $P \in I_F$ and μ is the empirical mean of all the points. The estimated covariance matrices are further used in order to characterize each texture image.

To speed-up the covariance matrices computation time, the fast covariance computation algorithm based on integral images, presented in [Tuzel *et al.* 2006], has been implemented. This procedure is described in Appendix D.

For the present application, patches of 15×15 pixels are extracted. In addition, an overlap of 8 pixels is considered between patches. Therefore, each image, in the VisTex and Outex database, is represented by a set of 36 and respectively 196 patches P, which will give a set of 36, respectively 196, covariance matrices of size $m \times m$, with $m = 5$. In the end, each texture class is represented by a set $\mathbf{M}_1, \dots, \mathbf{M}_N$ of N covariance matrices, with $\mathbf{M}_n \in \mathcal{P}_5$.

6.4.2.2 Codebook Creation

Knowing that supervised classification methods are considered later, the covariance matrices database is equally and randomly divided in order to obtain the training and the testing sets. Further on, the patches in the training set are used to create a codebook. For this step, a within-class approach is implemented. More precisely, each texture class is modeled by a mixture of K Riemannian distributions (RGD or RLD) and the estimated parameters $\{\hat{\varpi}_k, \widehat{\mathbf{M}}_k, \hat{\sigma}_k\}_{1 \leq k \leq K}$ represent the codewords.

The codebook is obtained by concatenating the codewords previously extracted for each class. The estimation procedure is carried out here, by using the intrinsic k-means algorithm detailed in Section 4.3.4.1 and Section 4.4.4.1, with K being set to 3. In addition, the k-means algorithm is repeated 10 times in order to reduce the influence of the centroids initialization.

6.4.2.3 Coding and Post-processing

Once that the codebook is determined, the extracted features are projected into the codebook space during the coding stage. The BoRW, RFV and R-VLAD models are derived for both RGDs and RLDs as explained in Section 6.2 and Section 6.3. After their computation, a post-processing step is required.

In the FV framework, the post-processing consists in two possible normalization steps [Faraki *et al.* 2014]:

- ℓ_2 *normalization* that has been proposed in [Perronnin *et al.* 2010b] to minimize the influence of the background information on the image signature. For a vector $\mathbf{V}$, its normalized version $\mathbf{V}_{L_2}$ is computed as:

$$\mathbf{V}_{L_2} = \frac{\mathbf{V}}{\|\mathbf{V}\|_2}, \tag{6.31}$$

  where $\|\cdot\|$ represents the L_2 norm.

- *power normalization* that corrects the independence assumption made on the patches [Perronnin *et al.* 2010a]. For the same vector $\mathbf{V}$, the power-normalized version $\mathbf{V}_{power}$ is obtained as:

$$\mathbf{V}_{power} = \text{sign}(\mathbf{V})|\mathbf{V}|^{\rho}, \tag{6.32}$$

  where $0 < \rho \leq 1$, and $\text{sign}(\cdot)$ is the signum function. For all the experiments presented in this section, ρ is set to $\frac{1}{2}$, as suggested in [Sánchez *et al.* 2013].

The same normalization scheme is applied for R-VLAD model. For the BoRW model, only ℓ_2 normalization is performed, as recommended in [Faraki *et al.* 2015b].

6.4.2.4 Classification Methods

For the final classification step, each test image is associated to the class of the most similar training image by using several approaches. For the first one, the support vector machine (SVM) [Vapnik 1995] algorithm with a Gaussian kernel is considered. In this case, the dispersion parameter in the Gaussian kernel is optimized by considering a cross-validation procedure on the training set. The practical implementation is made by using the LIBSVM library [Chang & Lin 2011]. For the second approach, the random forest classifier [Breiman 2001], with 100 trees, is applied for the RFV and R-VLAD descriptors and the results are compared to those given by the SVM.

6.4.3 Results

In this section, the classification results obtained on the VisTex and Outex_TC000_13 databases are discussed. Table 6.1 and Table 6.2 report the SVM classification performances in terms of overall accuracy. In order to find these values, the databases have been partitioned 10 times in training and testing sets. In addition, the Fisher information matrix given in (6.2) is considered to be the identity matrix.

In these tables, the first column specifies the descriptor's type (BoRW, RFV, or R-VLAD). The second column (Homosced.) refers to the homoscedasticity assumption. If this assumption is true, all the clusters c_k have the same dispersion parameter σ_k. The third column (Prior) corresponds to the weights ϖ_k. If this parameter is set to false, the same weight is given to all the clusters of the mixture model. The last two columns present the classification performances when mixtures of RGDs and RLDs model the space of estimated covariance matrices. Moreover, in the section concerning the BoRW, the results obtained by using the state-of-the-art method, described in [Faraki *et al.* 2014], are reported on the third row. The other lines refer to a modified version of this algorithm, implying the maximization of the RGD, or RLD, likelihood in the codebook creation.

The carried out experiments have multiple purposes. First, the RGD's and RLD's performances are analyzed, in order to discover the most suitable distribution for data modeling. Second, the descriptors are compared to find the most accurate one for the present problem. Third, for the RFV, the contribution of each parameter (weight, dispersion, centroid) to the classification accuracy is tested. For example, the row "RFV : ϖ" indicates the classification results when only the derivatives with respect to the weights are considered to calculate the RFV.

By observing the results, the following conclusions can be noticed. First, for these experiments, the use of RLDs brings little improvement in terms of classification accuracy. The most important raises can be spotted for the VisTex and Outex database by considering the "RFV: σ" (about 7%) and the "RFV: σ, ϖ" (about 4%) features. In both tables, the corresponding values are marked in blue. Moreover, combining the RFV associated to the centroid $\bar{\mathbf{M}}$ with those associated to the weight and dispersion parameters yields a gain of about 3% on the VisTex database for both RGDs and RLDs. In addition, the proposed RFV outperforms significantly the state-of-the-art BoRW [Faraki *et al.* 2014] and R-VLAD descriptors [Faraki *et al.* 2015a] . A significant gain of 3 to 4% is observed on these databases and the best classification results are marked in red. This gain is quite logical, since the RFV can be interpreted as a generalization of R-VLAD.

Next, the influence of the final classifier is analyzed for the RFV and R-VLAD. Therefore, the SVM and the random forest classification algorithms are tested on the VisTex and Outex databases. The results are presented in Table 6.3 and Table 6.4, knowing that the texture classes are modeled by a mixture of K RGDs. From this experiment, it can be concluded that the R-VLAD method is not influenced by the classification algorithms. On the other hand, for the RFV the best results are always obtained when the SVM classifier is used. An important gain of 4 to 6% can be

Table 6.1: SVM classification results on the VisTex database in terms of overall accuracy.

Method	Homosced.	Prior	RGD	RLD
BoRW	false	true	87.22 ± 1.19	87.70 ± 1.75
BoRW	false	false	87.51 ± 0.92	88.10 ± 1.42
BoRW [Faraki *et al.* 2014]	true	false	87.20 ± 1.55	87.69 ± 0.93
BoRW	true	true	76.67 ± 2.35	69.01 ± 5.39
RFV : ϖ	false	true	89.21 ± 0.94	90.11 ± 0.58
RFV : σ	false	true	81.42 ± 1.12	88.51 ± 0.87
RFV : $\bar{\mathbf{M}}$	false	true	87.22 ± 1.15	87.71 ± 1.06
RFV : σ, ϖ	false	true	81.80 ± 0.60	85.36 ± 0.86
RFV : $\bar{\mathbf{M}}, \varpi$	false	true	88.13 ± 0.67	88.45 ± 0.79
RFV : $\bar{\mathbf{M}}, \sigma$	false	true	90.41 ± 0.86	91.07 ± 0.53
RFV : $\bar{\mathbf{M}}, \sigma, \varpi$	false	true	89.93 ± 0.53	89.77 ± 1.13
R-VLAD [Faraki *et al.* 2015a]	true	false	87.94 ± 0.58	87.38 ± 0.73

Table 6.2: SVM classification results on the Outex database in terms of overall accuracy.

Method	Homosced.	Prior	RGD	RLD
BoRW	false	true	84.32 ± 0.99	83.84 ± 0.81
BoRW	false	false	84.37 ± 1.28	83.79 ± 0.96
BoRW [Faraki *et al.* 2014]	true	false	84.43 ± 1.23	83.60 ± 0.79
BoRW	true	true	79.31 ± 1.86	77.19 ± 0.27
RFV : ϖ	false	true	84.31 ± 1.29	84.32 ± 0.85
RFV : σ	false	true	78.46 ± 1.54	84.15 ± 1.01
RFV : $\bar{\mathbf{M}}$	false	true	83.94 ± 0.90	83.78 ± 0.67
RFV : σ, ϖ	false	true	79.72 ± 2.09	81.79 ± 0.92
RFV : $\bar{\mathbf{M}}, \varpi$	false	true	84.51 ± 0.78	84.40 ± 0.99
RFV : $\bar{\mathbf{M}}, \sigma$	false	true	84.32 ± 1.19	84.78 ± 1.11
RFV : $\bar{\mathbf{M}}, \sigma, \varpi$	false	true	84.57 ± 1.24	83.94 ± 1.23
R-VLAD [Faraki *et al.* 2015a]	true	false	82.99 ± 1.19	83.71 ± 1.32

observed for the VisTex database. It has to be mentioned that this comparison do not concern the BoRW, due to the fact that in this case, the image signatures are represented by histograms and not by some feature vectors as in the case of RFV and R-VLAD.

Table 6.3: Comparison between the SVM and the random forest classification performances on the VisTex database, for the RGD model.

Method	Homosced.	Prior	SVM	Random forest
RFV : $\bar{\mathbf{M}}$	false	true	87.22 ± 1.15	83.73 ± 1.18
RFV : $\bar{\mathbf{M}}, \sigma$	false	true	90.41 ± 0.86	85.15 ± 0.64
RFV : $\bar{\mathbf{M}}, \sigma, \varpi$	false	true	89.93 ± 0.53	85.03 ± 0.46
R-VLAD [Faraki *et al.* 2015a]	true	false	87.94 ± 0.58	87.97 ± 0.67

Table 6.4: Comparison between the SVM and the random forest classification performances on the Outex database for the RGD model.

Method	Homosced.	Prior	SVM	Random forest
RFV : $\bar{\mathbf{M}}$	false	true	83.94 ± 0.90	81.68 ± 1.33
RFV : $\bar{\mathbf{M}}, \sigma$	false	true	84.32 ± 1.19	81.82 ± 0.39
RFV : $\bar{\mathbf{M}}, \sigma, \varpi$	false	true	84.57 ± 1.24	81.63 ± 0.59
R-VLAD [Faraki *et al.* 2015a]	true	false	82.99 ± 1.19	83.93 ± 0.83

6.5 Conclusions and Perspectives

6.5.1 Conclusions

Starting from a generative model, local descriptors, such as Fisher vectors can be extracted in order to describe the information lying in signals, images, or videos. These FV are descriptors derived from the Fisher kernels and they represent a method to measure if samples are correctly fitted by a given model.

Introduced initially in the context of Gaussian mixture models [Perronnin & Dance 2007], FV have been generalized in this chapter to Riemannian manifolds, where the features are represented by parametric descriptors, like covariance matrices. The obtained descriptors have been called Riemannian Fisher vectors. Several aspects concerning this new local model have been presented in this chapter.

First, based on the definition of the mixtures of Riemannian Gaussian distributions and Riemannian Laplace distributions, the expressions of RFV have been developed. Knowing that these mixture models are characterized by three parameters, namely the mixture weight, the central value and the dispersion, the corresponding Fisher scores have been expressed. By concatenating some, or all of them, the proposed RFV have been obtained [Ilea *et al.* 2016b, Ilea *et al.* 2016a].

Second, the connection between the RFV and the Riemannian version of the conventional vectors of locally aggregated descriptors (R-VLAD) [Faraki *et al.* 2015a] has been analyzed. It has been shown that by considering the homoscedasticity hypothesis, along with a hard assignment scheme, the RFV reduces to R-VLAD. The proposed RFV can hence be considered as a generalization of R-VLAD descriptors.

Next, both Gaussian and Laplace RFV models have been applied in the context of texture image classification. In addition, their behavior has been compared to other local descriptors, already generalized for the Riemannian case, that are the bag of Riemannian words (BoRWs) [Faraki *et al.* 2014] and R-VLAD [Faraki *et al.* 2015a].

6.5.2 Perspectives

Further works will include several directions:

- *The derivation of an analytical expression of the Fisher information matrix for the Riemannian Gaussian and Laplace distributions:* in Section 6.2.1.2, it has been shown that the Fisher vectors are computed based on the gradient of the model's log-likelihood. The obtained expression is often normalized by the Fisher information matrix, which has a favorable impact on the results [Perronnin & Dance 2007]. Even though explicit forms can be derived for this matrix in the Euclidean space, up to our knowledge, there is no closed-form expression of the Fisher information matrix for the considered Riemannian Gaussian and Laplace distributions. In practice, it can be estimated by Monte Carlo integration. Knowing that in the Euclidean space, the results are improved after the normalization, further works will concern the search of an analytical expression of the Fisher information matrix for the Riemannian Gaussian and Laplace distributions.

- *The use of enhanced image descriptors:* the classification performances reported in this chapter have been obtained starting from basic descriptors. More precisely, for each image, the intensity and the norms of the first and second order derivatives have been considered in order to build the covariance matrices in Section 6.4.2.1. The impact of more complex descriptors, like the so-called local extrema-based descriptor (LED) [Pham *et al.* 2016] that captures all the color, spatial and gradient information will be analyzed in future works.

- *The dictionary reduction:* in this chapter, dictionary based classification methods that operate in the Riemannian space have been used. Further on, based on the recent works on sparse representation for symmetric positive definite matrices [Harandi *et al.* 2012,Harandi *et al.* 2016], the codebook creation stage described in Section 6.4.2.2 will be modified in order to take into consideration only some representative codewords. The obtained results will be integrated in the proposed classification workflow for the Riemannian distributions.

- *The exploitation of the spatial distribution of the extracted patches:* in this chapter, a patch based classification methods has been presented. While the proposed method has demonstrate promising results, it does not take into account the spatial distribution of the patches. Exploiting this information may yield to gain of classification performance. Inspired by the concept of co-occurrence matrices [Haralick *et al.* 1973], future works will concern the development of a classification method which will exploit the statistical dependence between neighboring patches.

Chapter 7

Conclusions and Perspectives

Contents

7.1 Conclusions

The work presented in this thesis focused on the use of covariance matrices, as texture descriptors, for the development of robust classification algorithms. More precisely, starting from the zero-mean multivariate Gaussian distributions, as statistical models for texture information, a robust classification workflow has been proposed. This workflow has been introduced in Chapter 1 and it is recalled in Figure 7.1, in order to summarize the main contributions of this work.

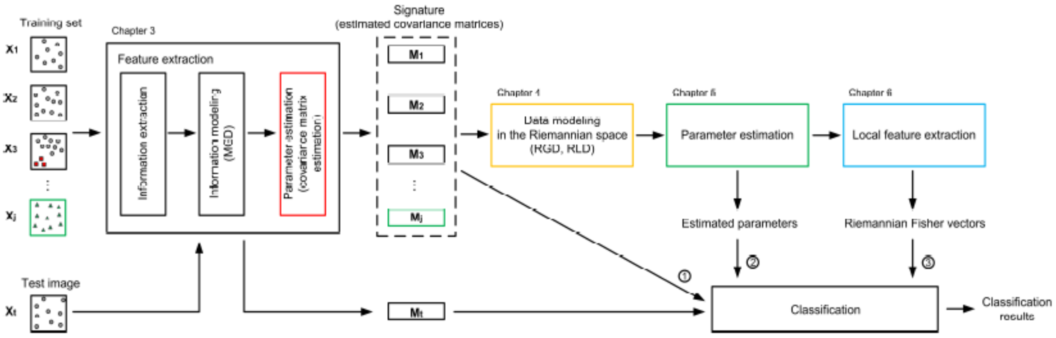


Figure 7.1: Classification workflow.

This workflow begins with the extraction of some textural descriptors and their modeling by means of zero-mean multivariate Gaussian distributions. Considering this starting point, the concept of robustness appears at different levels.

First, the estimation of the covariance matrix characterizing the zero-mean MGD has to be capable to deal with the outlier values present in the observations. Once extracted, these features can be used in the classification. At this stage, for the decision-making strategy, a method to regulate the false alarm rate is desired. To answer this need, a statistical hypothesis test has been proposed in this work, based on the geodesic distance. This test has been analyzed in terms of noise robustness and classification performance. The test statistic has been used further, for PolSAR image classification. In this context, the classification workflow has been modified by adding an optional preprocessing stage, consisting in noise filtering. Based on the partial differential equation formalism, a directional diffusion denoising method has been proposed, for speckle reduction. By applying this step, it has been shown that the overall accuracy can be improved.

Instead of directly using the covariance matrices in the classification, they can be modeled as elements on the Riemannian manifold, in order to take into consideration the geometry of their space. To model the within-class diversity, the Riemannian Gaussian and Laplace distributions have been introduced. The probability density functions of these distributions have two parameters: the central value and the dispersion around it. As a result, the model's robustness to outliers is given by the centroid estimation procedure. This remark has motivated the proposition of the Riemannian Laplace distribution to minimize the influence of outlier covariance matrices on the centroid estimation. For this model, the MLE of the central value is the median, which is known to be a robust estimator, compared to the center of mass used for the Riemannian Gaussian distribution. In addition, the mixture model has been also defined and their parameters have been estimated by extending the k-means and EM algorithms. The BIC criterion has been considered, for the automatic computation of the appropriate number of clusters involved in those two clustering algorithms. In the end, these methods have been applied to texture image classification and shown promising results compared to the state-of-the-art Wishart distribution.

The k-means and EM clustering algorithms are based on the partition of the dataset in subsets, characterized by their central value. A robust alternative to the center of mass and the median, previously mentioned, has been introduced in this thesis. This centroid has been inspired by the theory of M-estimators, more precisely by the Huber's estimator. The proposed estimator, called the Huber's centroid, represents a trade-off between the efficiency of the center of mass and the robustness of the median. This compromise is tuned by a threshold, which controls the estimator's behavior. More precisely, for large values, the centroid behaves as the Riemannian center of mass, while for small values it is similar to the median. To compute the estimated centroid, a gradient based algorithm has been also proposed. In addition, an algorithm for the automatic computation of the Huber's threshold has been defined, based on the concept of median absolute deviation, that has been extended to the Riemannian manifold. This estimator has been validated on simulated data, texture images, simulated PolSAR data and magnetoencephalography data. The results have shown its potential.

All the previous classification algorithms are based on global descriptors. The last contribution of this thesis, gives an alternative to this approach by defining a local descriptor based classification method. In this context, the Fisher vectors have been extended to the Riemannian manifold. This extension was not possible without a probabilistic generative model. By considering the Riemannian Gaussian and Laplace models, introduced in this thesis, the Riemannian Fisher vectors have been defined and the relation with the Riemannian vectors of locally aggregated descriptors has been shown. In the end, these descriptors have been applied to texture image classification, showing promising results.

7.2 Perspectives

Ideas related to the future work have been presented at the end of each chapter. In the following, a selection of the possible perspectives is made, representing the main research directions. Therefore, the work presented in this thesis can be continued, by considering the following:

- *The generalization of the proposed methods to non-Gaussian statistical models:* all the methods presented in this thesis rely on the use of covariance matrices, as texture features. A multivariate Gaussian distribution has been considered to model the observations. This choice has been motivated by the convenient properties of this distribution. More precisely, for the MGDs, the geodesic distance admits a closed form. In Chapter 2, it has been shown that more complex statistical models can be employed for data modeling, such as the multivariate generalized Gaussian distribution (MGGD), the SIRV model, the copula-based model, etc. Such models are expected to better represent the observations, since they all generalize the multivariate Gaussian distribution. But in general, for these models, the geodesic distance does not admit a closed form. Only some approximations can be available for some particular cases (fixed shape parameter, linear approximation of the geodesics). One may consider a divergence, like the Kullback-Leibler, instead of the geodesic distance but in this case, some desirable properties of the distance will be lost. Therefore, the adaptation of the proposed approaches to other probability models should be taken into consideration for future works.

- *The extension of the proposed Riemannian tools to the space of complex covariance matrices:* in Chapter 4, two density models for the space of real covariance matrices have been proposed, that are the Riemannian Gaussian and Laplace distributions. Future works will address the extension of these models to the space of complex covariance matrices. This idea is motivated by the existence of applications where the covariance matrices can be complex valued. For instance, this is the case of PolSAR data. This topic is the subject of current research works [Hajri *et al.* 2017].

- *The development of the Riemannian models for structured covariance matrices:* another extension of the Riemannian Gaussian and Laplace distribution is represented by the adaptation of these models to the space of structured covariance matrices. In practice, covariance matrices having special forms, like Tœplitz, or block-Tœplitz, can be found. Hence, the Riemannian distance on the manifold of structured covariance matrices is not the one defined in (4.5) and considered in this thesis. More precisely, for these matrices, the geometry of their space has to be respected by the chosen distance, in order to benefit of all their advantages. Consequently, all the developed Riemannian-geometric tools should be readapted to these spaces of structured covariance matrices [Said *et al.* 2016].

- *The exploitation of the spatial distribution of patches in the classification:* in Chapter 6, a patch based classification method has been introduced. Based on the concept of Fisher vectors extended to covariance matrices, the proposed Riemannian Fisher vectors have shown promising results, compared to state-of-the-art local covariance matrice descriptors, i.e. BoRW and R-VLAD. Nevertheless, all these methods have a major drawback: they do not exploit the spatial distribution of patches. Inspired by the concept of gray level co-occurrence matrices used for texture analysis, further works will be devoted to the extension of such co-occurrence matrix to covariance matrices estimated on local patches.

Appendix A

Creating Outlier Images for the PolSARproSim Database

In the context of forest stands, the outliers can be represented by stands with modified structure. The changes in structure can be determined by storms, illnesses, or human actions and they are reflected in the textural and polarimetric characteristics of the corresponding images. Therefore, by having properties that are different from the rest of the images in the dataset, they can be considered as oultiers.

In order to mimic this behavior, the simulated images are modified by replacing a predefined percentage of pixels with aberrant ones. The aberrant pixels are generated as vegetation pixels and structured in a circular area as shown in Figure A.1. The area's surface is computed by taking into consideration the percentage of aberrant pixels fixed by the user.

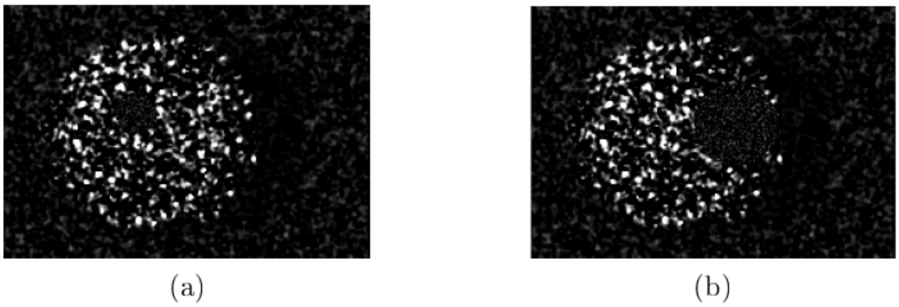


Figure A.1: Example of PolSARproSim outlier images containing (a) 5% and (b) 20% of aberrant pixels.

To obtain the outlier images, several steps are needed:

1. A PolSARproSim image with large surface and low tree density is simulated;

2. A large surface containing no trees is identified and modeled by a zero-mean multivariate Gaussian distribution.

3. The covariance matrix of the MGD for the selected region is estimated;

4. A new dataset having the same distribution, that is the same parameter value, is generated. The new dataset is structured in a circular area and inserted into another PolSARproSim image that will represent the outlier image:

(a) a circular area having a specific surface and the center randomly fixed is generated. The surface is computed based on the percentage of aberrant pixels fixed by the user.

(b) the pixels of the circular area are replaced by the earlier obtained MGD dataset.

Appendix B

Fisher Vectors for the Riemannian Gaussian Model

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be an N-sample of i.i.d observations modeled as a mixture of K Riemannian Gaussian distributions. Under the independence assumption, the probability density function of $\boldsymbol{\mathcal{M}}$ is given by:

$$p(\boldsymbol{\mathcal{M}}|\theta) = \prod_{n=1}^{N} p(\mathbf{M}_n|\theta), \tag{B.1}$$

where $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1\leq k \leq K}\}$ is the parameter vector containing the mixture weight ϖ_k, the central value $\bar{\mathbf{M}}_k$ and the dispersion parameter σ_k. For a mixture of K RGDs, $p(\mathbf{M}_n|\theta)$ is defined as [Said *et al.* 2015b]:

$$\begin{aligned} p(\mathbf{M}_n|\theta) &= \sum_{k=1}^{K} \varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\ &= \sum_{k=1}^{K} \varpi_k \; \frac{1}{Z(\sigma_k)} \exp\Big\{ - \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\}, \end{aligned} \tag{B.2}$$

where $Z(\sigma_k)$ is a normalization factor independent of the centroid $\bar{\mathbf{M}}_k$ and $d(\,\cdot\,)$ is the Riemannian distance [James 1973].

Starting from (B.1), the log-likelihood is obtained as:

$$\log p(\boldsymbol{\mathcal{M}}|\theta) = \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \tag{B.3}$$

and its derivatives with respect to the parameter vector can be computed.

B.1 The derivative with respect to the centroid $\bar{\mathbf{M}}_k$

The derivative of the log-likelihood in (B.3) with respect to the centroid $\bar{\mathbf{M}}_k$ is obtained as follows:

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} &= \frac{\partial}{\partial \bar{\mathbf{M}}_k} \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \\
&= \sum_{n=1}^{N} \frac{\partial \log p(\mathbf{M}_n|\theta)}{\partial \bar{\mathbf{M}}_k} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta)}{p(\mathbf{M}_n|\theta)} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta)}{\sum_{j=1}^{K} \varpi_j \, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}.
\end{aligned} \tag{B.4}$$

The numerator is separately computed further:

$$\begin{aligned}
\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta) &= \frac{\partial}{\partial \bar{\mathbf{M}}_k} \sum_{j=1}^{K} \varpi_j \, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \varpi_k \, \frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&= \varpi_k \, \frac{\partial}{\partial \bar{\mathbf{M}}_k} \frac{1}{Z(\sigma_k)} \exp\Big\{ - \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \\
&= \varpi_k \, \frac{1}{Z(\sigma_k)} \exp\Big\{ - \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \Big(- \frac{1}{2\sigma_k^2} \Big) \frac{\partial}{\partial \bar{\mathbf{M}}_k} d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k).
\end{aligned} \tag{B.5}$$

Knowing that derivative of the Riemannian distance is given by [Chavel 2006]:

$$\frac{\partial}{\partial \bar{\mathbf{M}}_k} d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k) = -2 \, \mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n), \tag{B.6}$$

the expression in (B.5) can be written:

$$\begin{aligned}
\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta) &= \varpi_k \, \frac{1}{Z(\sigma_k)} \exp\Big\{ - \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \Big(- \frac{1}{2\sigma_k^2} \Big) (-2 \, \mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)) \\
&= \varpi_k \, \frac{1}{Z(\sigma_k)} \exp\Big\{ - \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \frac{\mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)}{\sigma_k^2} \\
&= \varpi_k \, \frac{1}{\sigma_k^2} \, \mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n) \, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k).
\end{aligned} \tag{B.7}$$

Next, by replacing the numerator in (B.4) by the result in (B.7), the following relation is derived:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \frac{\frac{\varpi_k}{\sigma_k^2} \, \mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n) \, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \varpi_j \, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \tag{B.8}$$

In the end, by introducing the variable:

$$\gamma_k(\mathbf{M}_n) = \frac{\varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}, \tag{B.9}$$

the final form of the derivative with respect to $\bar{\mathbf{M}}_k$ is obtained:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \; \sigma_k^{-2} \; \mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n). \tag{B.10}$$

B.2 The derivative with respect to the dispersion σ_k

The derivative of the log-likelihood in (B.3) with respect to the dispersion parameter σ_k is obtained as follows:

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} &= \frac{\partial}{\partial \sigma_k} \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \\
&= \sum_{n=1}^{N} \frac{\partial \log p(\mathbf{M}_n|\theta)}{\partial \sigma_k} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\theta)}{p(\mathbf{M}_n|\theta)} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\theta)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}.
\end{aligned} \tag{B.11}$$

Next, the numerator is separately computed:

$$\begin{aligned}
\frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\theta) &= \frac{\partial}{\partial \sigma_k} \sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \varpi_k \; \frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&= \varpi_k \; \frac{\partial}{\partial \sigma_k} \frac{1}{Z(\sigma_k)} \exp\Big\{ -\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \\
&= \varpi_k \; \Bigg\{ -\frac{Z'(\sigma_k)}{Z^2(\sigma_k)} \exp\Big\{ -\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} + \\
&+ \frac{1}{Z(\sigma_k)} \exp\Big\{ -\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \Bigg\} \\
&= \varpi_k \; \frac{1}{Z(\sigma_k)} \exp\Big\{ -\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \Bigg\{ -\frac{Z'(\sigma_k)}{Z(\sigma_k)} + \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \Bigg\} \\
&= \varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \Bigg\{ -\frac{Z'(\sigma_k)}{Z(\sigma_k)} + \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \Bigg\}.
\end{aligned} \tag{B.12}$$

By replacing in (B.11) the numerator with the previously expression (B.12), the derivative of the log-likelihood becomes:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \frac{\varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)\Big\{ - \frac{Z'(\sigma_k)}{Z(\sigma_k)} + \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \Big\}}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \tag{B.13}$$

In the end, by using the expression of $\gamma_k(\mathbf{M}_n)$ in (B.9), the final form of the derivative with respect to σ_k is obtained:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n)\Big\{ - \frac{Z'(\sigma_k)}{Z(\sigma_k)} + \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \Big\}. \tag{B.14}$$

B.3 The derivative with respect to the weight ϖ_k

In order to compute the derivative of the log-likelihood in (B.3) with respect to the weight, a parametrization is needed first [Sánchez *et al.* 2013]:

$$\varpi_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{K} \exp(\alpha_j)}. \tag{B.15}$$

This parametrization using α_k ensures the constraints of positivity and sum to one for the weights. Therefore, the derivative with respect to ϖ_k is replaced by the computation of the derivative with respect to α_k:

$$\begin{aligned} \frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \alpha_k} &= \frac{\partial}{\partial \alpha_k} \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \\ &= \sum_{n=1}^{N} \frac{\partial \log p(\mathbf{M}_n|\theta)}{\partial \alpha_k} \\ &= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \alpha_k} p(\mathbf{M}_n|\theta)}{p(\mathbf{M}_n|\theta)} \\ &= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \alpha_k} p(\mathbf{M}_n|\theta)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \end{aligned} \tag{B.16}$$

The numerator is separately computed, by taking into consideration the parametrization in (6.18):

$$\begin{aligned}
\frac{\partial}{\partial \alpha_k} p(\mathbf{M}_n|\theta) &= \frac{\partial}{\partial \alpha_k} \sum_{j=1}^{K} \frac{\exp(\alpha_j)}{\sum_{l=1}^{K} \exp(\alpha_l)}\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \frac{\exp(\alpha_k) \sum_{l=1}^{K} \exp(\alpha_l) - \exp(\alpha_k)^2}{\left[\sum_{l=1}^{K} \exp(\alpha_l)\right]^2}\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&\quad + \sum_{j \neq k} \exp(\alpha_j)\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)\, \frac{-\exp(\alpha_k)}{\left[\sum_{l=1}^{K} \exp(\alpha_l)\right]^2} \\
&= (\varpi_k - \varpi_k^2)\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) - \sum_{j \neq k} \varpi_j\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)\, \varpi_k \\
&= (\varpi_k - \varpi_k^2)\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) + \varpi_k^2\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&\quad - \sum_{j=1}^{K} \varpi_j\, \varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) - \varpi_k \sum_{j=1}^{K} \varpi_j\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j). \qquad \text{(B.17)}
\end{aligned}$$

Next, by replacing the numerator in (B.16), by the previously obtained expression (B.17), the derivative of the log-lihelikood can be written as:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \frac{\varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) - \varpi_k\, \sum_{j=1}^{K} \varpi_j\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}{\sum_{j=1}^{K} \varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \qquad \text{(B.18)}$$

In the end, by introducing the expression of $\gamma_k(\mathbf{M}_n)$ in (B.9), the final form of the derivative with respect to α_k is obtained:

$$\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{M}_n) - \varpi_k\right]. \qquad \text{(B.19)}$$

Appendix C

Fisher Vectors for the Riemannian Laplace Model

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be an N-sample of i.i.d observations modeled as a mixture of K Riemannian Laplace distributions. Under the independence assumption, the probability density function of $\boldsymbol{\mathcal{M}}$ is given by:

$$p(\boldsymbol{\mathcal{M}}|\theta) = \prod_{n=1}^{N} p(\mathbf{M}_n|\theta), \tag{C.1}$$

where $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1\leq k \leq K}\}$ is the parameter vector containing the mixture weight ϖ_k, the central value $\bar{\mathbf{M}}_k$ and the dispersion parameter σ_k. For a mixture of K RLDs, $p(\mathbf{M}_n|\theta)$ is defined as [Hajri *et al.* 2016]:

$$\begin{aligned} p(\mathbf{M}_n|\theta) &= \sum_{k=1}^{K} \varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\ &= \sum_{k=1}^{K} \varpi_k \, \frac{1}{\zeta_m(\sigma_k)} \exp\Big\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\}, \end{aligned} \tag{C.2}$$

where $\zeta_m(\sigma_k)$ is a normalizing constant independent of $\bar{\mathbf{M}}_k$ and $d(\,\cdot\,)$ is the Riemannian distance [James 1973].

Starting from (C.1), the log-likelihood is obtained as:

$$\log p(\boldsymbol{\mathcal{M}}|\theta) = \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \tag{C.3}$$

and its derivatives with respect to the parameter vector can be computed.

C.1 The derivative with respect to the centroid $\bar{\mathbf{M}}_k$

The derivative of the log-likelihood in (C.3) with respect to the centroid $\bar{\mathbf{M}}_k$ is obtained as follows:

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} &= \frac{\partial}{\partial \bar{\mathbf{M}}_k} \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \\
&= \sum_{n=1}^{N} \frac{\partial \log p(\mathbf{M}_n|\theta)}{\partial \bar{\mathbf{M}}_k} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta)}{p(\mathbf{M}_n|\theta)} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}.
\end{aligned} \tag{C.4}$$

The numerator is separately computed further:

$$\begin{aligned}
\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta) &= \frac{\partial}{\partial \bar{\mathbf{M}}_k} \sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \varpi_k \; \frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&= \varpi_k \; \frac{\partial}{\partial \bar{\mathbf{M}}_k} \frac{1}{\zeta(\sigma_k)} \exp\Big\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \\
&= \varpi_k \; \frac{1}{\zeta(\sigma_k)} \exp\Big\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \Big(- \frac{1}{2\sigma_k^2} \Big) \frac{\partial}{\partial \bar{\mathbf{M}}_k} d(\mathbf{M}_n, \bar{\mathbf{M}}_k).
\end{aligned} \tag{C.5}$$

Knowing that derivative of the Riemannian distance is given by [Chavel 2006]:

$$\frac{\partial}{\partial \bar{\mathbf{M}}_k} d(\mathbf{M}_n, \bar{\mathbf{M}}_k) = - \frac{\mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)}{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}, \tag{C.6}$$

the expression in (C.5) can be written:

$$\begin{aligned}
\frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}_n|\theta) &= \varpi_k \; \frac{1}{\zeta(\sigma_k)} \exp\Big\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \Big(- \frac{1}{2\sigma_k^2} \Big) \left(- \frac{\mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)}{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)} \right) \\
&= \frac{\varpi_k}{2\sigma_k^2} \; \frac{1}{\zeta(\sigma_k)} \exp\Big\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \Big\} \left(\frac{\mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)}{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)} \right) \\
&= \frac{\varpi_k}{2\sigma_k^2} \; \frac{\mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)}{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)} \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k).
\end{aligned} \tag{C.7}$$

Next, by replacing the numerator in (C.4) by the result in (C.7), the following relation is derived:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \frac{\frac{\varpi_k}{2\sigma_k^2} \; \frac{\mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)}{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)} \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \tag{C.8}$$

In the end, by introducing the variable:

$$\gamma_k(\mathbf{M}_n) = \frac{\varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}, \tag{C.9}$$

the final form of the derivative with respect to $\bar{\mathbf{M}}_k$ is obtained:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \; \frac{\mathrm{Log}_{\bar{\mathbf{M}}_i}(\mathbf{M}_n)}{\sigma_k^2 \; d(\mathbf{M}_n, \bar{\mathbf{M}}_k)} \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k). \tag{C.10}$$

C.2 The derivative with respect to the dispersion σ_k

The derivative of the log-likelihood in (C.3) with respect to the dispersion parameter σ_k is obtained as follows:

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} &= \frac{\partial}{\partial \sigma_k} \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \\
&= \sum_{n=1}^{N} \frac{\partial \log p(\mathbf{M}_n|\theta)}{\partial \sigma_k} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\theta)}{p(\mathbf{M}_n|\theta)} \\
&= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\theta)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}.
\end{aligned} \tag{C.11}$$

Next, the numerator is separately computed:

$$\begin{aligned}
\frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\theta) &= \frac{\partial}{\partial \sigma_k} \sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \varpi_k \; \frac{\partial}{\partial \sigma_k} p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&= \varpi_k \; \frac{\partial}{\partial \sigma_k} \frac{1}{\zeta(\sigma_k)} \exp\left\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \right\} \\
&= \varpi_k \left\{ - \frac{\zeta'(\sigma_k)}{\zeta^2(\sigma_k)} \exp\left\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \right\} + \right. \\
&\left. + \frac{1}{\zeta(\sigma_k)} \exp\left\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \right\} \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \right\} \\
&= \varpi_k \; \frac{1}{\zeta(\sigma_k)} \exp\left\{ - \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{2\sigma_k^2} \right\} \left\{ - \frac{\zeta'(\sigma_k)}{\zeta(\sigma_k)} + \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \right\} \\
&= \varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \left\{ - \frac{\zeta'(\sigma_k)}{\zeta(\sigma_k)} + \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \right\}.
\end{aligned} \tag{C.12}$$

By replacing in (C.11) the numerator with the previously expression (C.12), the derivative of the log-likelihood becomes:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \frac{\varpi_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)\Big\{ -\frac{\zeta'(\sigma_k)}{\zeta(\sigma_k)} + \frac{d(\mathbf{M}_n,\bar{\mathbf{M}}_k)}{\sigma_k^3} \Big\}}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \tag{C.13}$$

In the end, by using the expression of $\gamma_k(\mathbf{M}_n)$ in (C.9), the final form of the derivative with respect to σ_k is obtained:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n)\Big\{ -\frac{\zeta'(\sigma_k)}{\zeta(\sigma_k)} + \frac{d(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} \Big\}. \tag{C.14}$$

C.3 The derivative with respect to the weight ϖ_k

In order to compute the derivative of the log-likelihood in (C.3) with respect to the weight, a parametrization is needed first [Sánchez *et al.* 2013]:

$$\varpi_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{K} \exp(\alpha_j)}. \tag{C.15}$$

This parametrization using α_k ensures the constraints of positivity and sum to one for the weights. Therefore, the derivative with respect to ϖ_k is replaced by the computation of the derivative with respect to α_k:

$$\begin{aligned} \frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \alpha_k} &= \frac{\partial}{\partial \alpha_k} \sum_{n=1}^{N} \log p(\mathbf{M}_n|\theta) \\ &= \sum_{n=1}^{N} \frac{\partial \log p(\mathbf{M}_n|\theta)}{\partial \alpha_k} \\ &= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \alpha_k} p(\mathbf{M}_n|\theta)}{p(\mathbf{M}_n|\theta)} \\ &= \sum_{n=1}^{N} \frac{\frac{\partial}{\partial \alpha_k} p(\mathbf{M}_n|\theta)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \end{aligned} \tag{C.16}$$

The numerator is separately computed, by taking into consideration the parametrization in (6.18):

$$
\begin{aligned}
\frac{\partial}{\partial \alpha_k} p(\mathbf{M}_n|\theta) &= \frac{\partial}{\partial \alpha_k} \sum_{j=1}^{K} \frac{\exp(\alpha_j)}{\sum_{l=1}^{K} \exp(\alpha_l)}\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \frac{\exp(\alpha_k) \sum_{l=1}^{K} \exp(\alpha_l) - \exp(\alpha_k)^2}{\left[\sum_{l=1}^{K} \exp(\alpha_l)\right]^2}\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&\quad + \sum_{j \neq k} \exp(\alpha_j)\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)\, \frac{-\exp(\alpha_k)}{\left[\sum_{l=1}^{K} \exp(\alpha_l)\right]^2} \\
&= (\varpi_k - \varpi_k^2)\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) - \sum_{j \neq k} \varpi_j\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)\, \varpi_k \\
&= (\varpi_k - \varpi_k^2)\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) + \varpi_k^2\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) \\
&\quad - \sum_{j=1}^{K} \varpi_j\, \varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j) \\
&= \varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) - \varpi_k \sum_{j=1}^{K} \varpi_j\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j). \qquad \text{(C.17)}
\end{aligned}
$$

Next, by replacing the numerator in (C.16), by the previously obtained expression (C.17), the derivative of the log-lihelikood can be written as:

$$
\frac{\partial \log p(\boldsymbol{\mathcal{M}}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \frac{\varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k) - \varpi_k \sum_{j=1}^{K} \varpi_j\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}{\sum_{j=1}^{K} \varpi_k\, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \qquad \text{(C.18)}
$$

In the end, by introducing the expression of $\gamma_k(\mathbf{M}_n)$ in (C.9), the final form of the derivative with respect to α_k is obtained:

$$
\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{M}_n) - \varpi_k\right]. \qquad \text{(C.19)}
$$

Appendix D

Integral Images for Covariance Matrix Computation

Integral images are intermediate image representations that have been introduced in [Viola & Jones 2001]. Starting from an image I, the integral image I_I is defined as:

$$I_I(x', y') = \sum_{x<x',\, y<y'} I(x, y). \tag{D.1}$$

A graphical representation of this equation is shown in Figure D.1. Moreover, the definition of integral images can be extended to higher dimensions and used for fast computation of region sums, as mentioned in [Tuzel *et al.* 2006]. In the same work, the procedure for covariance matrix estimation based on integral images has been detailed, as follows. First, the expression of the (i, j)-th element in the covariance matrix given in (6.30), has been rewritten:

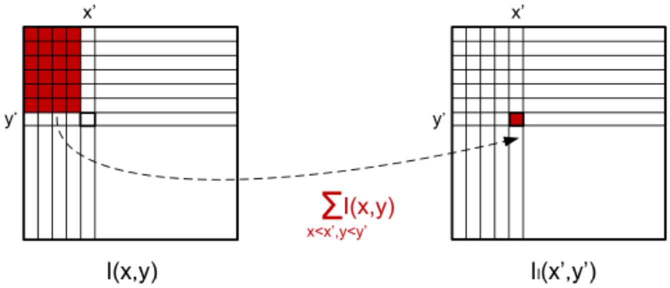


Figure D.1: Computation of integral images.

$$\begin{aligned}\mathbf{M}_P(i, j) &= \frac{1}{N_P}\sum_{n=1}^{N_P}\Big(\mathbf{p}_n(i) - \mu(i)\Big)\Big(\mathbf{p}_n(j) - \mu(j)\Big) = \\ &= \frac{1}{N_P}\left(\sum_{n=1}^{N_P}\mathbf{p}_n(i)\mathbf{p}_n(j) - \frac{1}{N_P}\sum_{n=1}^{N_P}\mathbf{p}_n(i)\sum_{n=1}^{N_P}\mathbf{p}_n(j)\right).\end{aligned} \tag{D.2}$$

To compute the two terms in (D.2), the integral images have been used. Therefore, the following notations have been introduced:

$$R(x', y', i) = \sum_{x<x',\, y<y'} I_F(x, y, i) \tag{D.3}$$

and

$$Q(x',y',i,j) = \sum_{x<x',\, y<y'} I_F(x,y,i,j), \tag{D.4}$$

where R is the $W \times H \times m$ tensor of the integral images, Q is the $W \times H \times m \times m$ tensor of the second order integral images and $i, j = 1, \ldots, m$. In addition,

$$\mathbf{R}_{x,y} = \begin{bmatrix} R(x,y,1) & \ldots & R(x,y,m) \end{bmatrix}^T \tag{D.5}$$

and

$$\mathbf{Q}_{x,y} = \begin{bmatrix} Q(x,y,1,1) & \ldots & Q(x,y,1,m) \\ & \vdots & \\ Q(x,y,m,1) & \ldots & Q(x,y,m,m) \end{bmatrix} \tag{D.6}$$

are the corresponding m-dimensional vector and $m \times m$ dimensional matrix. Finally, the covariance matrix of a patch P delimited by the upper left corner (x',y') and the lower right (x'',y'') corner, as illustrated in Figure D.2, is given by:

$$\begin{aligned} \mathbf{M}_{P(x',y';x'',y'')} = \frac{1}{N_P}\Big(&\mathbf{Q}_{x'',y''} + \mathbf{Q}_{x',y'} - \mathbf{Q}_{x'',y'} - \mathbf{Q}_{x',y''} - \\ -\frac{1}{N_P}\big(\mathbf{R}_{x'',y''} + \mathbf{R}_{x',y'} - \mathbf{R}_{x',y''} - \mathbf{R}_{x'',y'}\big)&\big(\mathbf{R}_{x'',y''} + \mathbf{R}_{x',y'} - \mathbf{R}_{x',y''} - \mathbf{R}_{x'',y'}\big)^T\Big), \end{aligned} \tag{D.7}$$

knowing that $N_P = (x'' - x')(y'' - y')$. This expression is obtained based on the fact that the sum of the elements in any rectangle (x',y',x'',y'') contained in an integer image I_I can be evaluated as:

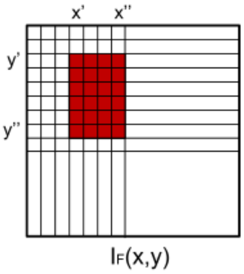


Figure D.2: Example of a patch $P \in I_F$, where each element p_n is an m-dimensional point, $n = 1, \ldots, N_P$.

$$I_I(x'',y'') + I_I(x',y') - I_I(x'',y') - I_I(x',y''). \tag{D.8}$$

With this algorithm, once that the integral images have been built, the time complexity of the covariance matrix estimation is $O(m^2)$, $\forall P \in I_F$ [Tuzel *et al.* 2006].

Bibliography

[Absil *et al.* 2008] P.-A. Absil, R. Mahony and R. Sepulchre. Optimization algorithms on matrix manifolds. Princeton University Press, Princeton, NJ, 2008. (Cited on page 75.)

[Afsari 2011] B. Afsari. *Riemannian Lp center of mass: existence, uniqueness and convexity.* Proceedings of the American Mathematical Society, vol. 139, no. 2, pages 655–673, 2011. (Cited on pages 18, 52, 62 and 74.)

[Amadasun & King 1989] M. Amadasun and R. King. *Textural features corresponding to textural properties.* IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, no. 5, pages 1264–1274, 1989. (Cited on page 9.)

[Anderson 1984] T.W. Anderson. An introduction to multivariate statistical analysis. Wiley Series in Probability and Statistics. Wiley, 1984. (Cited on page 22.)

[Armijo 1966] L. Armijo. *Minimization of functions having Lipschitz continuous first partial derivatives.* Pacific Journal of Mathematics, vol. 16, no. 1, pages 1–3, 1966. (Cited on page 75.)

[Arnaudon *et al.* 2013] M. Arnaudon, F. Barbaresco and L. Yang. Medians and means in Riemannian geometry: Existence, uniqueness and computation, pages 169–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. (Cited on page 71.)

[Aujol *et al.* 2003] J.-F. Aujol, G. Aubert and L. Blanc-Feraud. *Wavelet-based level set evolution for classification of textured images.* IEEE Transactions on Image Processing, vol. 12, no. 12, pages 1634–1641, 2003. (Cited on page 20.)

[Barachant *et al.* 2012] A. Barachant, S. Bonnet, M. Congedo and C. Jutten. *Multiclass brain-computer interface classification by Riemannian geometry.* IEEE Transactions on Biomedical Engineering, vol. 59, no. 4, pages 920–928, 2012. (Cited on pages 69 and 92.)

[Barachant *et al.* 2013] A. Barachant, S. Bonnet, M. Congedo and C. Jutten. *Classification of covariance matrices using a Riemannian-based kernel for BCI applications.* NeuroComputing, vol. 112, pages 172–178, 2013. (Cited on pages 2, 16, 52 and 100.)

[Barachant 2014] A. Barachant. MEG *decoding using* Riemannian *geometry and unsupervised classification.* Technical Report, 2014. (Cited on page 91.)

[Barbaresco *et al.* 2013] F. Barbaresco, M. Arnaudon and L. Yang. *Riemannian medians and means with applications to Radar signal processing.* IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 4, pages 595–604, 2013. (Cited on pages 2, 16, 52, 74, 81 and 97.)

[Bishop 2007] C. M. Bishop. Pattern recognition and machine learning (information science and statistics). Springer, 1st ed. 2006. corr. 2nd printing 2011 édition, 2007. (Cited on pages 18, 52, 62 and 74.)

[Bombrun & Beaulieu 2008] L. Bombrun and J.-M. Beaulieu. *Fisher distribution for texture modeling of polarimetric sar data*. IEEE Geoscience and Remote Sensing Letters, vol. 5, no. 3, pages 512–516, 2008. (Cited on pages 16, 17 and 52.)

[Bombrun *et al.* 2011a] L. Bombrun, S. N. Anfinsen and O. Harant. *A complete coverage of log-cumulant space in terms of distributions for polarimetric SAR data*. In 5th International Workshop on Science and Applications of SAR Polarimetry and Polarimetric Interferometry (POLinSAR 2011), pages 1–8, 2011. (Cited on pages 16 and 17.)

[Bombrun *et al.* 2011b] L. Bombrun, Y. Berthoumieu, N.-E. Lasmar and G. Verdoolaege. *Mutlivariate texture retrieval using the geodesic distance between elliptically distributed random variables*. In IEEE International Conference on Image Processing, pages 3637–3640, 2011. (Cited on pages 15 and 20.)

[Boubchir *et al.* 2010] L. Boubchir, A. Nait-Ali and E. Petit. *Multivariate statistical modeling of images in sparse multiscale transforms domain*. In 2010 IEEE International Conference on Image Processing, pages 1877–1880, 2010. (Cited on pages 12 and 14.)

[Breiman 2001] L. Breiman. *Random forests*. Machine Learning, vol. 45, no. 1, pages 5–32, 2001. (Cited on page 113.)

[Candes & Donoho 1999] E. J. Candes and D. L. Donoho. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Technical report. Department of Statistics, Stanford University, 1999. (Cited on page 12.)

[Chang & Lin 2011] C.-C. Chang and C.-J. Lin. *LIBSVM: A library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, vol. 2, pages 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. (Cited on page 113.)

[Chavel 2006] I. Chavel. Riemannian geometry: A modern introduction. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006. (Cited on pages 126 and 132.)

[Chen *et al.* 2011] Y. Chen, A. Wiesel and A. O. Hero. *Robust shrinkage estimation of high-dimensional covariance matrices*. IEEE Transactions on Signal Processing, vol. 59, no. 9, pages 4097–4107, Sept 2011. (Cited on pages 2 and 16.)

[Conte *et al.* 2002] E. Conte, A. De Maio and G. Ricci. *Recursive estimation of the covariance matrix of a compound-Gaussian process and its application to adaptive CFAR detection*. IEEE Transactions on Signal Processing, vol. 50, no. 8, pages 1908–1915, 2002. (Cited on page 23.)

[CRP 2008] *Le pin maritime: pilier de l'économie forestière d'Aquitaine*. Technical report, Centre Régional de la Propriété Forestière d'Aquitaine, 2008. (Cited on page 31.)

[Csurka *et al.* 2004] G. Csurka, C. R. Dance, L. Fan, J. Willamowski and C. Bray. *Visual categorization with bags of keypoints*. In Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision, pages 1–22, 2004. (Cited on pages 102 and 103.)

[Curran 1988] P. J. Curran. *The semivariogram in remote sensing: An introduction*. Remote Sensing of Environment, vol. 24, no. 3, pages 493 – 507, 1988. (Cited on pages 9 and 11.)

[Daubechies 1992] I. Daubechies. Ten lectures on wavelets. Society for Industrial and Applied Mathematics, 1992. (Cited on page 13.)

[de Luis-García *et al.* 2011] R. de Luis-García, C-F. Westin and C. Alberola-López. *Gaussian mixtures on tensor fields for segmentation: applications to medical imaging*. Computerized Medical Imaging and Graphics, vol. 35, no. 1, pages 16–30, 2011. (Cited on pages 2 and 16.)

[Dec] *DecMeg2014 - Decoding the Human Brain*. Biomag 2014 Decoding Challenge: Brain Decoding Across Subjects. Available: https://www.kaggle.com/c/decoding-the-human-brain. (Cited on page 91.)

[Del] *NL–SAR Toolbox*. Non-Local framework for (Pol)(In)SAR denoising. Available: https://www.math.u-bordeaux.fr/∼cdeledal/nlsar.php. (Cited on page 44.)

[Deledalle *et al.* 2015] C. A. Deledalle, L. Denis, F. Tupin, A. Reigber and M. Jäger. *NL-SAR: A Unified Nonlocal Framework for Resolution-Preserving (Pol)(In)SAR Denoising*. IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 4, pages 2021–2038, 2015. (Cited on pages 40, 44 and 45.)

[Di Zenzo 1986] S. Di Zenzo. *A Note on the Gradient of a Multi-image*. Computer Vision, Graphics, and Image Processing, vol. 33, no. 1, pages 116–125, 1986. (Cited on page 41.)

[Do & Vetterli 2002] M. N. Do and M. Vetterli. *Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance*. IEEE Transactions on Image Processing, vol. 11, pages 146–158, 2002. (Cited on pages 14 and 20.)

[Donoho 1995] D. L. Donoho. *Denoising by soft-thresholding*. IEEE Transactions on Information Theory, vol. 41, no. 3, pages 613–627, 1995. (Cited on page 20.)

[Douze *et al.* 2011] M. Douze, A. Ramisa and C. Schmid. *Combining attributes and Fisher vectors for efficient image retrieval*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 745–752, 2011. (Cited on page 100.)

[Faraki *et al.* 2014] M. Faraki, M. T. Harandi, A. Wiliem and B. C. Lovell. *Fisher tensors for classifying human epithelial cells*. Pattern Recognition, vol. 47, no. 7, pages 2348 – 2359, 2014. (Cited on pages 3, 6, 100, 102, 106, 107, 111, 113, 114, 115 and 116.)

[Faraki *et al.* 2015a] M. Faraki, M. T. Harandi and F. Porikli. *More about VLAD: A leap from Euclidean to Riemannian manifolds*. In IEEE Conference on Computer Vision and Pattern Recognition, 2015, pages 4951–4960, June 2015. (Cited on pages 2, 3, 6, 16, 59, 100, 102, 107, 110, 111, 114, 115 and 116.)

[Faraki *et al.* 2015b] M. Faraki, M. Palhang and C. Sanderson. *Log-Euclidean bag of words for human action recognition*. IET Computer Vision, vol. 9, no. 3, pages 331–339, 2015. (Cited on pages 6, 100, 102, 106, 107 and 113.)

[Fiori 2009] S. Fiori. *Learning the Fréchet mean over the manifold of symmetric positive-definite matrices*. Cognitive Computation, vol. 1, no. 4, pages 279–291, 2009. (Cited on pages 75 and 76.)

[Fletcher & Joshi 2007] P. T. Fletcher and S. Joshi. *Riemannian geometry for the statistical analysis of diffusion tensor data*. Signal Processing, vol. 87, no. 2, pages 250 – 262, 2007. (Cited on page 54.)

[Fletcher *et al.* 2009] P. T. Fletcher, S. Venkatasubramanian and S. Joshi. *The geometric median on Riemannian manifolds with application to robust atlas estimation*. Neuroimage, vol. 45, no. 1, pages S143–S152, 2009. (Cited on pages 52, 55, 64, 74, 75, 76, 77 and 81.)

[Formont *et al.* 2011] P. Formont, F. Pascal, G. Vasile, J. Ovarlez and L. Ferro-Famil. *Statistical classification for heterogeneous polarimetric SAR images*. IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 3, pages 567–576, 2011. (Cited on pages 2 and 16.)

[Formont *et al.* 2013] P. Formont, J.-P. Ovarlez and F. Pascal. *On the use of matrix information geometry for polarimetric SAR image classification*. In F. Nielsen and R. Bhatia, editors, Matrix Information Geometry, pages 257–276. Springer Berlin Heidelberg, 2013. (Cited on pages 18, 52, 62 and 74.)

[Formont 2013] P. Formont. *Outils statistiques et géométriques pour la classification des images SAR polarimétriques hautement texturées*. PhD thesis, Université Rennes 1, 2013. (Cited on page 95.)

[Freitas *et al.* 2003] C. C. Freitas, A. C. Frery, A. H. Correia, R. C. Frery, C. A. C. Sim oes and A. Brazil. *The polarimetric G distribution for SAR data analysis*, 2003. (Cited on pages 16, 17 and 52.)

[Garcia & Oller 2006] G. Garcia and J. M. Oller. *What does intrinsic mean in statistical estimation?* Statistics and Operations Research Transactions, vol. 30, no. 2, pages 125–170, 2006. (Cited on pages 87 and 100.)

[Georgeson 1979] M. A. Georgeson. *Spatial Fourier analysis and human vision*. Tutorial essays in psychology, vol. 2, pages 39–88, 1979. (Cited on page 12.)

[Gini & Greco 2002] F. Gini and M. V. Greco. *Covariance matrix estimation for CFAR detection in correlated heavy tailed clutter*. Signal Processing, vol. 82, no. 12, pages 1847–1859, 2002. (Cited on pages 15 and 23.)

[Goodman 1963] N. R. Goodman. *Statistical analysis based on a certain multivariate complex Gaussian distribution (An introduction)*. Annals of Mathematical Statistics, vol. 34, no. 1, pages 152–177, 03 1963. (Cited on page 52.)

[Greco *et al.* 2014] M. Greco, S. Fortunati and F. Gini. *Maximum likelihood covariance matrix estimation for complex elliptically symmetric distributions under mismatched conditions*. Signal Process., vol. 104, pages 381–386, 2014. (Cited on pages 2 and 16.)

[Gu *et al.* 2014] X. Gu, J. D. Deng and M. K. Purvis. *Improving superpixel-based image segmentation by incorporating color covariance matrix manifolds*. In IEEE International Conference on Image Processing, pages 4403–4406, 2014. (Cited on page 97.)

[Haar 1910] A. Haar. *Zur Theorie der orthogonalen Funktionensysteme*. Mathematische Annalen, vol. 69, no. 3, pages 331–371, 1910. (Cited on page 13.)

[Hajri *et al.* 2016] H. Hajri, I. Ilea, S. Said, L. Bombrun and Y. Berthoumieu. *Riemannian Laplace distribution on the space of symmetric positive definite matrices*. Entropy, vol. 18, no. 3, page 98, 2016. (Cited on pages 16, 18, 52, 62, 63, 64, 65, 69, 100, 131 and 157.)

[Hajri *et al.* 2017] H. Hajri, S. Said, L. Bombrun and Y. Berthoumieu. *A geometric learning approach on the space of complex covariance matrices*. In submitted to IEEE International Conference on Acoustics, Speech, and Signal Processing, 2017. (Cited on pages 71 and 122.)

[Hampel *et al.* 2005] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw and W. A. Stahel. Robust statistics. John Wiley & Sons, Inc., 2005. (Cited on page 98.)

[Haralick *et al.* 1973] R. M. Haralick, K. Shanmugam and I. Dinstein. *Textural features for image classification*. IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-3, no. 6, pages 610–621, 1973. (Cited on pages 9, 10, 33 and 117.)

[Harandi *et al.* 2012] M. T. Harandi, C. Sanderson, R. Hartley and B. C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach, pages 216–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. (Cited on page 117.)

[Harandi *et al.* 2016] M. T. Harandi, R. Hartley, B. Lovell and C. Sanderson. *Sparse coding on symmetric positive definite manifolds using bregman divergences*. IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 6, pages 1294–1306, 2016. (Cited on page 117.)

[Helgason 2001] S. Helgason. Differential geometry,Lie groups, and symmetric spaces. Crm Proceedings & Lecture Notes. American Mathematical Society, 2001. (Cited on page 54.)

[Henson *et al.* 2011] R. N. Henson, D. G. Wakeman, V. Litvak and K. J. Friston. *A Parametric Empirical Bayesian framework for the* EEG/MEG *inverse problem: generative models for multisubject and multimodal integration*. Frontiers in Human Neuroscience, vol. 5, no. 76, 2011. (Cited on page 91.)

[Higham 2008] N. J. Higham. Functions of matrices: Theory and computation. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. (Cited on pages 55, 75, 76, 77, 83, 107, 109 and 110.)

[Huber & Ronchetti 2009] P. J. Huber and E. M. Ronchetti. Robust statistics. John Wiley & Sons, Inc., 2009. (Cited on page 85.)

[Huber 1964] P. J. Huber. *Robust estimation of a location parameter*. The Annals of Mathematical Statistics, vol. 35, no. 1, pages 73–101, 1964. (Cited on pages 21, 24, 25, 74, 81 and 82.)

[Ilea *et al.* 2015a] I. Ilea, L. Bombrun, C. Germain, R. Terebes and M. Borda. *Classification robuste sur l'espace des matrices de covariance : Application à l'imagerie polarimétrique radar*. In XXVème colloque GRETSI, 2015. (Cited on pages 21, 33 and 158.)

[Ilea *et al.* 2015b] I. Ilea, L. Bombrun, C. Germain, R. Terebes and M. Borda. *Statistical hypothesis test for robust classification on the space of covariance matrices*. In IEEE International Conference on Image Processing, pages 271–275, 2015. (Cited on pages 21, 28, 29, 30, 33, 100 and 157.)

[Ilea *et al.* 2015c] I. Ilea, L. Bombrun, G. Germain, I. Champion, R. Terebes and M. Borda. *Statistical hypothesis test for maritime pine forest SAR images* *classification based on the geodesic distance*. In IEEE International Geoscience and Remote Sensing Symposium, pages 3215–3218, 2015. (Cited on pages 21, 31, 32, 33, 34, 37, 157 and 171.)

[Ilea *et al.* 2016a] I. Ilea, L. Bombrun, C. Germain and Y. Berthoumieu. *Texture image classification with Riemannian Fisher vectors issued from a Laplacian model*. In 2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop, pages 1–5, 2016. (Cited on pages 110, 116 and 157.)

[Ilea *et al.* 2016b] I. Ilea, L. Bombrun, C. Germain, R. Terebes, M. Borda and Y. Berthoumieu. *Texture image classification with Riemannian Fisher vectors*. In IEEE International Conference on Image Processing, pages 3543 – 3547, 2016. (Cited on pages 109, 116, 157 and 165.)

[Ilea *et al.* 2016c] I. Ilea, L. Bombrun, R. Terebes, M. Borda and C. Germain. *An M-Estimator for robust centroid estimation on the manifold of covariance matrices*. IEEE Signal Processing Letters, vol. 23, no. 9, pages 1255–1259, 2016. (Cited on pages 74, 82, 85, 89, 157 and 159.)

[Ilea *et al.* 2016d] I. Ilea, H Hajri, S. Said, L. Bombrun, C. Germain and Y. Berthoumieu. *An M-estimator for robust centroid estimation on the manifold of covariance matrices: performance analysis and application to image classification*. In 24th European Signal Processing Conference, pages 2196 – 2200, 2016. (Cited on pages 74, 82 and 157.)

[Jaakkola & Haussler 1998] T. Jaakkola and D. Haussler. *Exploiting generative models in discriminative classifiers*. In In Advances in Neural Information Processing Systems 11, pages 487–493. MIT Press, 1998. (Cited on pages 100, 103 and 104.)

[Jain & Farrokhnia 1991] A. K. Jain and F. Farrokhnia. *Unsupervised texture segmentation using Gabor filters*. Pattern Recognition, vol. 24, no. 12, pages 1167–1186, 1991. (Cited on page 12.)

[Jakobsson *et al.* 2000] A. Jakobsson, S. L. Marple and P. Stoica. *Computationally efficient two-dimensional Capon spectrum analysis*. IEEE Transactions on Signal Processing, vol. 48, no. 9, pages 2651–2661, 2000. (Cited on page 72.)

[James 1973] A. T. James. The variance information manifold and the functions on it, pages 157 – 169. Academic Press, 1973. (Cited on pages 27, 54, 58, 64, 76, 107, 109, 110, 125 and 131.)

[Jeffreys 1946] H. Jeffreys. *An invariant form for the prior probability in estimation problems*. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, vol. 186, no. 1007, pages 453–461, 1946. (Cited on page 20.)

[Jégou *et al.* 2010] H. Jégou, M. Douze, C. Schmid and P. Pérez. *Aggregating local descriptors into a compact image representation.* In IEEE Conference on Computer Vision & Pattern Recognition, 2010. (Cited on pages 105 and 106.)

[Joachims 1998] T. Joachims. *Text categorization with suport vector machines: learning with many relevant features.* In Proceedings of the 10th European Conference on Machine Learning, pages 137–142. Springer-Verlag, 1998. (Cited on page 102.)

[Julesz *et al.* 1978] B. Julesz, E. N. Gilbert and J. D. Victor. *Visual discrimination of textures with identical third-order statistics.* Biological Cybernetics, vol. 31, no. 3, pages 137–140, 1978. (Cited on page 9.)

[Julesz 1962] B. Julesz. *Visual pattern discrimination.* IRE Transactions on Information Theory, vol. 8, no. 2, pages 84–92, 1962. (Cited on page 9.)

[Julesz 1981] B. Julesz. *Textons, the elements of texture perception, and their interactions.* Natura, vol. 290, no. 5802, pages 91–97, 1981. (Cited on page 9.)

[Julesz 1986] B. Julesz. *Texton gradients: The texton theory revisited.* Biological Cybernetics, vol. 54, no. 4, pages 245–251, 1986. (Cited on page 9.)

[Karcher 1977] H. Karcher. *Riemannian center of mass and mollifier smoothing.* Communications on Pure and Applied Mathematics, vol. 30, no. 5, pages 509–541, 1977. (Cited on pages 75 and 98.)

[Koller & Stahel 2011] M. Koller and W. A. Stahel. *Sharpening Wald-type inference in robust regression for small samples.* Computational Statistics and Data Analysis, vol. 55, no. 8, pages 2504–2515, 2011. (Cited on page 98.)

[Kullback & Leibler 1951] S. Kullback and R.A. Leibler. *On information and sufficiency.* The Annals of Mathematical Statistics, vol. 22, no. 1, pages 79–86, 1951. (Cited on pages 20 and 27.)

[Kupperman 1957] M. Kupperman. *Further applications of information theory to multivariate analysis and statistical inference.* PhD thesis, George Washington University, 1957. (Cited on pages 21 and 26.)

[Kwitt & Uhl 2010] R. Kwitt and A. Uhl. *Lightweight Probabilistic Texture Retrieval.* IEEE Transactions on Image Processing, vol. 19, no. 1, pages 241–253, 2010. (Cited on page 20.)

[Kwitt *et al.* 2009] R. Kwitt, P. Meerwald and A. Uhl. *A joint model of complex wavelet coefficients for texture retrieval.* 2009. (Cited on pages 14 and 15.)

[Landy & Graham 2004] M. S. Landy and N. Graham. *Visual perception of texture.* In The Visual Neurosciences, pages 1106–1118. MIT Press, 2004. (Cited on page 12.)

[Lasmar & Berthoumieu 2014] N. E. Lasmar and Y. Berthoumieu. *Gaussian copula multivariate modeling for texture image retrieval using wavelet transforms.* IEEE Transactions on Image Processing, vol. 23, no. 5, pages 2246–2261, 2014. (Cited on pages 14 and 15.)

[Lebedev & Silverman 1972] N. N. Lebedev and R. A. Silverman. Special functions and their applications. Dover Books on Mathematics. Dover Publications, 1972. (Cited on page 56.)

[Lee & Pottier 2009] J.-S. Lee and E. Pottier. Polarimetric radar imaging: from basics to applications. CRC Press, Taylor and Francis, 2009. (Cited on page 40.)

[Lee *et al.* 1993] J.-S. Lee, D. L. Schuler, L. H. Lang and K. J. Ranson. *K distribution for multilook processed polarimetric SAR imagery.* In IEEE International Geoscience and Remote Sensing Symposium, pages 2179–2181, 1993. (Cited on pages 16, 17 and 52.)

[Lee *et al.* 1999] J. S. Lee, M. R. Grunes, T. L. Ainsworth, L. J. Du, D. L. Schuler and S. R. Cloude. *Unsupervised classification using polarimetric decomposition and the complex Wishart classifier.* IEEE Transactions on Geoscience and Remote Sensing, vol. 37, no. 5, pages 2249–2258, 1999. (Cited on pages 17, 66 and 68.)

[Lee 1980] J.-S. Lee. *Digital image enhancement and noise filtering by use of local statistics.* Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 2, pages 165–168, 1980. (Cited on page 40.)

[Lee 1981] J. S. Lee. *Refined filtering of image noise using local statistics.* Computer Vision, Graphics, Image Processing, vol. 15, no. 2, 1981. (Cited on pages 40, 45 and 46.)

[Lloyd 2006] S. Lloyd. *Least squares quantization in PCM.* IEEE Transactions on Information Theory, vol. 28, no. 2, pages 129–137, 2006. (Cited on page 74.)

[MacQueen 1967] J. MacQueen. *Some methods for classification and analysis of multivariate observations.* In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pages 281–297, Berkeley, Calif., 1967. University of California Press. (Cited on page 74.)

[Mader & Reese 2012] K. Mader and G. Reese. *Using covariance matrices as feature descriptors for vehicle detection from a fixed camera.* ArXiv e-prints, February 2012. (Cited on pages 2, 16 and 100.)

[Mahot *et al.* 2013] M. Mahot, F. Pascal, P. Forster and J. P. Ovarlez. *Asymptotic Properties of Robust Complex Covariance Matrix Estimates.* IEEE Transactions on Signal Processing, vol. 61, no. 13, pages 3348–3356, 2013. (Cited on pages 16 and 25.)

[Mallat 1989] S. G. Mallat. *A theory for multiresolution signal decomposition: the wavelet representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pages 674–693, 1989. (Cited on page 12.)

[Manjunath & Ma 1996] B. S. Manjunath and W. Y. Ma. *Texture features for browsing and retrieval of image data*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pages 837–842, 1996. (Cited on page 12.)

[Marĉelja 1980] S. Marĉelja. *Mathematical description of the responses of simple cortical cells*. Journal of the Optical Society of America, vol. 70, no. 11, pages 1297–1300, 1980. (Cited on page 12.)

[Maronna *et al.* 2006] R. A. Maronna, D. R. Martin and V. J. Yohai. Robust statistics: theory and methods. John Wiley and Sons, New York, 2006. (Cited on page 98.)

[Maronna 1976] R. A. Maronna. *Robust M-estimators of multivariate location and scatter*. Annals of Statistics, vol. 4, no. 1, pages 51–67, 1976. (Cited on pages 24, 81 and 82.)

[Matheron 1963] G. Matheron. Principles of geostatistics, volume 58. Society of Economic Geologists, 1963. (Cited on pages 9 and 11.)

[Mathiassen *et al.* 2002] J. R. Mathiassen, A. Skavhaug and K. Bø. *Texture similarity measure using Kullback-Leibler divergence between gamma distributions*. In Proceedings of the 7th European Conference on Computer Vision-Part III, pages 133–147, London, UK, UK, 2002. Springer-Verlag. (Cited on page 14.)

[Maugé 1987] J. P. Maugé. Le pin maritime premier résineux de france. Centre de Productivité et d'Action Forestière d'Aquitaine, Institut pour le Développement Forestier, Paris, 1987. (Cited on page 31.)

[Moakher 2006] M. Moakher. *On the averaging of symmetric positive-definite tensors*. Journal of Elasticity, vol. 82, no. 3, pages 273–296, 2006. (Cited on page 76.)

[Moon & Stirling 2000] T. K. Moon and W. C. Stirling. Mathematical methods and algorithms for signal processing. Prentice Hall, Upper Saddle River, NJ, 2000. (Cited on page 25.)

[Mora *et al.* 2012] M. Mora, F. Córdova-Lepe and R. Del-Valle. *A non-Newtonian gradient for contour detection in images with multiplicative noise*. Pattern Recognition Letters, vol. 33, no. 10, pages 1245–1256, 2012. (Cited on pages 40 and 41.)

[Muirhead 1982] R. J. Muirhead. Aspects of multivariate statistical theory. Wiley Series in Probability and Statistics. Wiley, 1982. (Cited on pages 56 and 63.)

[Nascimento *et al.* 2010] A. D. C. Nascimento, R. J. Cintra and A. C. Frery. *Hypothesis testing in speckled data with stochastic distances*. IEEE Transactions on Geoscience and Remote Sensing, vol. 48, pages 373–385, 2010. (Cited on pages 21 and 26.)

[Nielsen & Bhatia 2012] F. Nielsen and R. Bhatia. Matrix information geometry. Springer Berlin Heidelberg, 2012. (Cited on pages 75 and 76.)

[Nielsen 2013] F. Nielsen. Pattern learning and recognition on statistical manifolds: An information-geometric review, pages 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. (Cited on page 52.)

[Ojala *et al.* 1996] T. Ojala, M. Pietikäinen and D. Harwood. *A comparative study of texture measures with classification based on feature distributions*. Pattern Recognition, vol. 29, no. 1, pages 51–59, 1996. (Cited on pages 9 and 11.)

[Ojala *et al.* 2002] T. Ojala, M. Pietikäinen and T. Mäenpää. *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pages 971–987, 2002. (Cited on page 11.)

[Olivetti *et al.* 2014] E. Olivetti, S. M. Kia and P. Avesani. *MEG decoding across subjects*. In International Workshop on Pattern Recognition in Neuroimaging, PRNI, 2014, Tübingen, Germany, 2014, pages 1–4, 2014. (Cited on page 91.)

[Oller & Corcuera 1995] J. M. Oller and J. M. Corcuera. *Intrinsic analysis of statistical estimation*. The Annals of Statistics, vol. 23, no. 5, pages 1562–1581, 1995. (Cited on page 87.)

[Ollila & Koivunen 2003] E. Ollila and V. Koivunen. *Robust antenna array processing using M-estimators of pseudo-covariance*. In 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, volume 3, pages 2659–2663, 2003. (Cited on pages 2, 16, 24 and 25.)

[Out] *Outex Texture Database*. Center for Machine Vision Research of the University of Oulu. Available: http://www.outex.oulu.fi/index.php?page=classification. (Cited on page 111.)

[Pascal *et al.* 2006] F. Pascal, J.-P. Ovarlez, P. Forster and P. Larzabal. *On a SIRV-CFAR detector with radar experimentations in impulsive clutter*. In 14th European Signal Processing Conference, volume 134, pages 1–5, 2006. (Cited on page 15.)

[Pascal *et al.* 2008] F. Pascal, P. Forster, J. Ovarlez and P. Larzabal. *Performance analysis of covariance matrix estimates in impulsive noise*. IEEE Transactions on Signal Processing, vol. 56, no. 6, pages 2206–2217, 2008. (Cited on page 23.)

[Pascal *et al.* 2013] F. Pascal, L. Bombrun, J.-Y. Tourneret and Y. Berthoumieu. *Parameter Estimation For Multivariate Generalized Gaussian Distributions*. IEEE Transactions on Signal Processing, vol. 61, no. 23, pages 5960–5971, December 2013. (Cited on page 15.)

[Pennec *et al.* 2006] X. Pennec, P. Fillard and N. Ayache. *A Riemannian framework for tensor computing*. International Journal of Computer Vision, vol. 66, no. 1, pages 41–66, 2006. (Cited on page 54.)

[Pennec 2006] X. Pennec. *Intrinsic statistics on Riemannianmanifolds: basic tools for geometric measurements*. Journal of Mathematical Imaging and Vision, vol. 25, no. 1, pages 127–154, 2006. (Cited on page 76.)

[Perronnin & Dance 2007] F. Perronnin and C. Dance. *Fisher kernels on visual vocabularies for image categorization*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, 2007. (Cited on pages 100, 103, 104, 108, 116 and 117.)

[Perronnin *et al.* 2010a] F. Perronnin, Y. Liu, J. Sánchez and H. Poirier. *Large-scale image retrieval with compressed Fisher vectors*. In The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pages 3384–3391, 2010. (Cited on pages 102 and 113.)

[Perronnin *et al.* 2010b] F. Perronnin, J. Sánchez and T. Mensink. Improving the Fisher kernel for large-scale image classification, volume 6314 of *Lecture Notes in Computer Science*, pages 143–156. Springer Berlin Heidelberg, 2010. (Cited on pages 102 and 113.)

[Pham *et al.* 2016] M.-T. Pham, G. Mercier, O. Regniers and J. Michel. *Texture retrieval from VHR optical remote sensed images using the local extrema descriptor with application to vineyard parcel detection*. Remote Sensing, vol. 8, no. 5, page 368, 2016. (Cited on page 117.)

[Pol] *PolSARpro*. Polarimetric SAR Data Processing and Educational Tool. Available: https://earth.esa.int/web/polsarpro/home. (Cited on page 45.)

[Prendes *et al.* 2015] J. Prendes, M. Chabert, F. Pascal, A. Giros and J. Y. Tourneret. *Change detection for optical and radar images using a Bayesian nonparametric model coupled with a Markov random field*. In IEEE International Conference on Acoustics, Speech and Signal Processing, pages 1513–1517, 2015. (Cited on page 60.)

[Rao & Lohse 1993] A. R. Rao and G. L. Lohse. *Identifying high level features of texture perception*. CVGIP: Graphical Models and Image Processing, vol. 55, no. 3, pages 218 – 233, 1993. (Cited on page 9.)

[Regniers *et al.* 2015a] O. Regniers, L. Bombrun, D. Guyon, J.-C. Samalens and C. Germain. *Wavelet-based texture features for the classification of age classes* *in a maritime pine forest*. IEEE Geosc. and Rem. Sens. Lett., vol. 12, no. 3, pages 621–625, 2015. (Cited on page 33.)

[Regniers *et al.* 2015b] O. Regniers, L. Bombrun, I. Ilea, V. Lafon and C. Germain. *Classification of oyster habitats by combining wavelet-based texture features and polarimetric SAR descriptors*. In IEEE International Geoscience and Remote Sensing Symposium, pages 3890–3893, 2015. (Cited on page 157.)

[Robinson 2005] J. Robinson. *Covariance matrix estimation for appearance-based face image processing*. Proceedings of the British Machine Vision Conference 2005, pages 389–398, 2005. (Cited on pages 2, 16 and 100.)

[Rosu *et al.* 2016] R. Rosu, M. Donias, L. Bombrun, S. Said, O. Regniers and J. P. Da Costa. *Structure tensor riemannian statistical models for CBIR and classification of remote sensing images*. IEEE Transactions on Geoscience and Remote Sensing, vol. PP, no. 99, pages 1–13, 2016. (Cited on page 54.)

[Ruch 2012] J.-J. Ruch. Statistique: Tests d'hypothèse. Préparation à l'Agrégation Bordeaux 1. 2012. (Cited on page 25.)

[Said *et al.* 2015a] S. Said, L. Bombrun and Y. Berthoumieu. *Texture classification using Rao's distance on the space of covariance matrices*. In Geometric Science of Information, 2015. (Cited on pages 2, 16, 54, 56, 58, 59, 74, 97 and 100.)

[Said *et al.* 2015b] S. Said, L. Bombrun, Y. Berthoumieu and J. H. Manton. *Riemannian Gaussian distributions on the space of symmetric positive definite matrices*. available on arxiv via http://arxiv.org/abs/1507.01760, 2015. (Cited on pages 16, 17, 52, 54, 55, 56, 58, 100 and 125.)

[Said *et al.* 2016] S. Said, H. Hajri, L. Bombrun and B. C. Vemuri. *Gaussian distributions on Riemannian symmetric spaces: statistical learning with structured covariance matrices*. ArXiv e-prints, 2016. (Cited on pages 72 and 122.)

[Saint-Jean & Nielsen 2013] C. Saint-Jean and F. Nielsen. A new implementation of k-mle for mixture modeling of wishart distributions, pages 249–256. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. (Cited on pages 17, 66 and 68.)

[Salicru *et al.* 1994] M. Salicru, D. Morales, M.L. Menendez and L. Pardo. *On the applications of divergence type measures in testing statistical hypotheses*. Journal of Multivariate Analysis, vol. 51, no. 2, pages 372 – 391, 1994. (Cited on pages 21 and 26.)

[Salton & Buckley 1988] G. Salton and C. Buckley. *Term-weighting approaches in automatic text retrieval*. Information Processing and Management, vol. 24, no. 5, pages 513–523, 1988. (Cited on pages 100 and 102.)

[Sánchez *et al.* 2013] J. Sánchez, F. Perronnin, T. Mensink and J. Verbeek. *Image classification with the Fisher vector: Theory and practice*. International Journal of Computer Vision, vol. 105, no. 3, pages 222–245, 2013. (Cited on pages 100, 103, 105, 106, 108, 113, 128 and 134.)

[Schwarz 1978] G. Schwarz. *Estimating the dimension of a model*. The Annals of Statistics, vol. 6, no. 2, pages 461–464, 1978. (Cited on page 60.)

[Smith 2005] S. T. Smith. *Covariance, subspace, and intrinsic Cramér-Rao bounds*. IEEE Transactions on Signal Processing, vol. 53, no. 5, pages 1610–1630, 2005. (Cited on page 87.)

[Srivastava *et al.* 2002] A. Srivastava, X. Liu and U. Grenander. *Universal analytical forms for modeling image probabilities*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 9, pages 1200–1214, 2002. (Cited on page 14.)

[Stitou *et al.* 2009] Y. Stitou, N.-E. Lasmar and Y. Berthoumieu. *Copulas based multivariate Gamma modeling for texture classification*. In IEEE International Conference on Acoustic Speech and Signal Processing, pages 1045–1048, 2009. (Cited on page 20.)

[Tamura *et al.* 1978] H. Tamura, S. Mori and T. Yamawaki. *Texture features corresponding to visual perception*. IEEE Transactions on System, Man and Cybernatic, vol. 6, 1978. (Cited on pages 9 and 12.)

[Terebes *et al.* 2004] R. Terebes, M. Borda, Y. Baozong, O. Lavialle and P. Baylou. *A new PDE based approach for image restoration and enhancement using robust diffusion directions and directional derivatives based diffusivities*. In Proceedings of the 7th International Conference on Signal Processing, volume 1, pages 707–712, 2004. (Cited on page 42.)

[Terebes *et al.* 2015] R. Terebes, M. Borda, C. Germain, R. Malutan and I. Ilea. *A multiplicative gradient-based anisotropic diffusion approach for speckle noise removal*. In E-Health and Bioengineering Conference, pages 1–6, 2015. (Cited on pages 21, 40, 41 and 158.)

[Terebes *et al.* 2016] R. Terebes, M. Borda, R. Malutan, C. Germain, L. Bombrun and I. Ilea. *PolSAR image denoising using directional diffusion*. accepted at International Symposium on Electronics and Telecommunications, 2016. (Cited on pages 21, 40, 44, 46 and 157.)

[Terras 1988] A. Terras. Harmonic analysis on symmetric spaces and applications. Number vol. 1 de Harmonic Analysis on Symmetric Spaces and Applications. Springer-Verlag, 1988. (Cited on page 54.)

[Therrien 1981] C. Therrien. *Relations between 2-D and multichannel linear prediction*. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 3, pages 454–456, 1981. (Cited on page 71.)

[Therrien 1992] C. Therrien. Discrete random signals and statistical signal processing. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st édition, 1992. (Cited on page 71.)

[Tsiotsios & Petrou 2013] C. Tsiotsios and M. Petrou. *On the choice of the parameters for anisotropic diffusion in image processing*. Pattern Recognition, vol. 46, no. 5, pages 1369–1381, 2013. (Cited on page 41.)

[Tuceryan & Jain 1993] M. Tuceryan and A. K. Jain. *The Handbook of Pattern Recognition and Computer Vision*. chapter Texture Analysis, pages 235–276. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1993. (Cited on pages 9 and 11.)

[Turner 1986] M. R. Turner. *Texture discrimination by Gabor functions*. Biological Cybernetics, vol. 55, no. 2-3, pages 71–82, November 1986. (Cited on page 12.)

[Tuzel *et al.* 2006] O. Tuzel, F. Porikli and P. Meer. Region covariance: A fast descriptor for detection and classification, volume 3952 of *Lecture Notes in Computer Science*, pages 589–600. Springer Berlin Heidelberg, 2006. (Cited on pages 111, 112, 137 and 138.)

[Tyler 1987] D. E. Tyler. *A distribution-free M-estimator of multivariate scatter*. The Annals of Statistics, vol. 15, no. 1, pages 234–251, 03 1987. (Cited on pages 21, 23, 24, 74 and 81.)

[Uehara *et al.* 2016] T. Uehara, T. Tanaka and S. Fiori. Robust averaging of covariance matrices by riemannian geometry for motor-imagery brain–computer interfacing, pages 347–353. Springer Singapore, Singapore, 2016. (Cited on pages 74, 75, 76, 78, 79, 81 and 92.)

[Vapnik 1995] V. N. Vapnik. The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA, 1995. (Cited on page 113.)

[Vasile *et al.* 2006] G. Vasile, E. Trouvé, J.-S. Lee and V. Buzuloiu. *Intensity-driven adaptive-neighborhood technique for polarimetric and interferometric SAR parameters estimation*. IEEE Transactions on Geoscience and Remote Sensing, vol. 44, pages 1609–1621, 2006. (Cited on pages 40, 45 and 46.)

[Vasile *et al.* 2010] G. Vasile, J. Ovarlez, F. Pascal and C. Tison. *Coherency matrix estimation of heterogeneous clutter in high-resolution polarimetric SAR images*. IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 4, pages 1809–1826, 2010. (Cited on pages 15 and 23.)

[Verdoolaege & Scheunders 2011] G. Verdoolaege and P. Scheunders. *Geodesics on the manifold of multivariate generalized Gaussian distributions with an application to multicomponent texture discrimination*. International Journal of Computer Vision, vol. 95, no. 3, pages 265–286, 2011. (Cited on pages 14 and 15.)

[Verdoolaege & Scheunders 2012] G. Verdoolaege and P. Scheunders. *On the geometry of multivariate generalized Gaussian models*. Journal of Mathematical Imaging and Vision, vol. 43, no. 3, pages 180–193, 2012. (Cited on page 20.)

[Viola & Jones 2001] P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages I–511–I–518, 2001. (Cited on page 137.)

[Vis] *Vision Texture Database*. MIT Vision and Modeling Group. Available: http://vismod.media.mit.edu/pub/VisTex. (Cited on pages 66, 89 and 111.)

[Wang & He 1990] L. Wang and D.-C. He. *Texture classification using texture spectrum*. Pattern Recognition, vol. 23, no. 8, pages 905–910, 1990. (Cited on page 11.)

[Weiszfeld 1937] E. Weiszfeld. *Sur le point pour lequel la somme des distances de n points donnés est minimum*. Tôhoku Mathematical Journal, vol. 43, pages 355–386, 1937. (Cited on page 77.)

[Whitaker 1993] R. T. Whitaker. *Geometry-Limited Diffusion*. PhD thesis, The University of North Carolina, Chapel Hill, North Carolina 27599-3175, 1993. (Cited on page 45.)

[Williams 2006] M. L. Williams. *PolSARproSim: A coherent, polarimetric SAR simulation of forests for PolSARPro. design document ans algorithm specification (u1.0)* . 2006. (Cited on pages 30 and 94.)

[Wishart 1928] J. Wishart. *The generalised product moment distribution in samples from a Normal multivariate population*. Biometrika, vol. 20A, no. 1/2, pages 32–52, 1928. (Cited on pages 16 and 52.)

[Yang *et al.* 2010] L. Yang, M. Arnaudon and F. Barbaresco. *Riemannian median, geometry of covariance matrices and radar target detection*. European Radar Conference, pages 415–418, 2010. (Cited on pages 2, 16, 75 and 77.)

[Yang 2010] L. Yang. *Riemannian median and its estimation*. LMS Journal of Computation and Mathematics, vol. 13, pages 461–479, 2010. (Cited on pages 52, 74, 76, 77, 78, 81 and 98.)

[Yao 1973] K. Yao. *A representation theorem and its applications to spherically-invariant random processes*. IEEE Transactions on Information Theory, vol. 19, no. 5, pages 600–608, 1973. cited By 194. (Cited on pages 14 and 15.)

[Yu & Acton 2002] Y. Yu and S. T. Acton. *Speckle reducing anisotropic diffusion*. IEEE Transactions on Image Processing, vol. 11, no. 11, pages 1260–1270, 2002. (Cited on pages 40, 44 and 45.)

[Zanini *et al.* 2016] P. Zanini, M. Congedo, C. Jutten, S. Said and Y. Berthoumieu. *Parameters estimate of Riemannian Gaussian distribution in the manifold of covariance matrices*. In IEEE Sensor Array and Multichannel Signal Processing Workshop, Rio de Janeiro, Brazil, 2016. (Cited on page 57.)

[Zhong *et al.* 2014] H. Zhong, J. Zhang and G. Liu. *Robust polarimetric SAR despeckling based on nonlocal means and distributed Lee filter*. IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 7, pages 4198–4210, 2014. (Cited on pages 40, 45 and 46.)

List of Publications

Journal Articles

1. I. Ilea, L. Bombrun, R. Terebes, M. Borda and C. Germain. *An M-Estimator for robust centroid estimation on the manifold of covariance matrices.* IEEE Signal Processing Letters, vol. 23, no. 9, pages 1255–1259, 2016
2. H. Hajri, I. Ilea, S. Said, L. Bombrun and Y. Berthoumieu. *Riemannian Laplace distribution on the space of symmetric positive definite matrices.* Entropy, vol. 18, no. 3, page 98, 2016

Conference Papers

1. I. Ilea, L. Bombrun, C. Germain, R. Terebes, M. Borda and Y. Berthoumieu. *Texture image classification with Riemannian Fisher vectors.* In IEEE International Conference on Image Processing, pages 3543 – 3547, 2016
2. I. Ilea, H Hajri, S. Said, L. Bombrun, C. Germain and Y. Berthoumieu. *An M-estimator for robust centroid estimation on the manifold of covariance matrices: performance analysis and application to image classification.* In 24th European Signal Processing Conference, pages 2196 – 2200, 2016
3. I. Ilea, L. Bombrun, C. Germain and Y. Berthoumieu. *Texture image classification with Riemannian Fisher vectors issued from a Laplacian model.* In 2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop, pages 1–5, 2016
4. R. Terebes, M. Borda, R. Malutan, C. Germain, L. Bombrun and I. Ilea. *PolSAR image denoising using directional diffusion.* accepted at International Symposium on Electronics and Telecommunications, 2016
5. I. Ilea, L. Bombrun, C. Germain, R. Terebes and M. Borda. *Statistical hypothesis test for robust classification on the space of covariance matrices.* In IEEE International Conference on Image Processing, pages 271–275, 2015
6. I. Ilea, L. Bombrun, G. Germain, I. Champion, R. Terebes and M. Borda. *Statistical hypothesis test for maritime pine forest SAR images classification based on the geodesic distance.* In IEEE International Geoscience and Remote Sensing Symposium, pages 3215–3218, 2015
7. O. Regniers, L. Bombrun, I. Ilea, V. Lafon and C. Germain. *Classification of oyster habitats by combining wavelet-based texture features and polarimetric SAR descriptors.* In IEEE International Geoscience and Remote Sensing Symposium, pages 3890–3893, 2015

8. I. Ilea, L. Bombrun, C. Germain, R. Terebes and M. Borda. *Classification robuste sur l'espace des matrices de covariance : Application à l'imagerie polarimétrique radar.* In XXVème colloque GRETSI, 2015

9. R. Terebes, M. Borda, C. Germain, R. Malutan and I. Ilea. *A multiplicative gradient-based anisotropic diffusion approach for speckle noise removal.* In E-Health and Bioengineering Conference, pages 1–6, 2015

An M-estimator for Robust Centroid Estimation on the Manifold of Covariance Matrices

I. Ilea, L. Bombrun, R. Terebes, M. Borda and C. Germain. *An M-Estimator for robust centroid estimation on the manifold of covariance matrices*. IEEE Signal Processing Letters, vol. 23, no. 9, pages 1255–1259, 2016

An M-estimator for Robust Centroid Estimation on the Manifold of Covariance Matrices

Ioana Ilea, Lionel Bombrun, Romulus Terebes, Monica Borda, and Christian Germain

***Abstract*—This paper introduces a new robust estimation method for the central value of a set of N covariance matrices. This estimator, called the Huber's centroid, is described starting from the expression of two well-known methods, that are the center of mass and the median. In addition, a computation algorithm based on the gradient descent is proposed. Moreover, the Huber's centroid performances are analyzed on simulated data, to identify the impact of outliers on the estimation process. In the end, the algorithm is applied to brain decoding, based on magnetoencephalography (MEG) data. For both simulated and real data, the covariance matrices are considered as realizations of Riemannian Gaussian distributions and the results are compared to those given by the center of mass and the median.**

***Index Terms*—centroid, classification, center of mass, median, Huber's centroid.**

I. Introduction

COVARIANCE matrices are used in a wide variety of applications in signal and image processing, including array processing [1], radar detection [2], [3], medical image segmentation [4], face detection [5], vehicle detection [6], etc. Another research direction concerns the signal and image classification, where covariance matrices can be used to model different kind of dependence, like spatial, temporal, spectral, polarimetric dependence, etc [7]–[10].

Recently, covariance matrices have been modeled as realizations of Riemannian Gaussian distributions (RGDs) and used in classification algorithms such as k-means or Expectation-Maximization (EM) [9]. This kind of classification procedures are based on the partition of the dataset in subsets, or clusters, characterized by their central values, also called centroids. The dataset's partition is accomplished by assigning each observation to the closest cluster in terms of a predefined distance [11]. This is a recursive procedure and for each iteration, the centroid's value is recomputed and the assignation step is repeated. Usually, the cluster's centroid is the center of mass, computed by using the squared Euclidean distance. Despite its popularity, this method is not appropriate for covariance matrices having a Riemannian geometry. To solve this problem, the Euclidean distance can be replaced by an intrinsic metric such as the Riemannian distance. The main disadvantage of the center of mass is its non-robust behavior to outliers that can exist in the dataset [11]–[13]. A robust alternative for the centroid's computation is the median, which has been also generalized for Riemannian manifolds [3], [14], [15]. This estimator is computed by using a gradient descent algorithm. Nonetheless, in this algorithm, a division by the distance between each observed covariance matrix in the dataset and the median is needed. If those two points are too close, this distance tends toward zero and may lead to numerical instability. In such case, Yang propose to exclude those points, at each iteration of the algorithm [14]. Another possibility for determining robust centroids in the space of covariance matrices is the use of the trimming methods [16]. These algorithms imply the elimination of a fixed percentage of outliers, according to their distance with respect to the dataset's mean or median, and the computation of the mean or the median on the remaining data. Nevertheless, the main difficulty of the trimmed estimators relies on the way to tune the percentage of discarded data.

The main contribution of the paper is to propose a novel centroid estimator, based on the theory of M-estimators. By considering the so-called Huber's function [17], [18], we introduce the definition of this estimator and present an algorithm to estimate it from a sample of N covariance matrices. The proposed estimator is a trade-off between the center of mass and the median, where the former is efficient, while the latter is robust to outliers. Moreover, based on the median abosolute deviation (MAD) concept, this paper presents a way to automatically determine the Huber's threshold.

The paper is structured as follows. Section II recalls the definition of the centroid from a sample of N observations. A brief overview of the center of mass and the median are given. Next, we introduce the proposed Huber's centroid estimator and present a gradient descent algorithm to estimate it. The performance of these estimators is then evaluated on simulated data. Section III introduces an application to brain decoding, based on MEG data. Finally, Section IV reports some conclusions and perspectives of this work.

II. The Huber's Estimator for Cluster Centroids

A. *Centroids and estimation methods*

Many signal and image processing applications including classification [9], segmentation [19], or filtering [3] require the computation of the central value of a covariance matrix dataset, which represents the subject of this section. Let $\{\mathrm{M}_1, \ldots, \mathrm{M}_N\}$ be a random sample of N covariance matrices. The centroid estimator of this set, denoted $\widehat{\mathrm{M}}$, is defined as being the minimizer of the following cost function $f(\mathrm{M})$:

$$\widehat{\mathrm{M}} = \underset{\mathrm{M}}{\operatorname{argmin}}\, f(\mathrm{M}). \qquad (1)$$

Depending on the choice of $f(\mathrm{M})$, different estimators of the centroids have been introduced in the literature. In the

I. Ilea, L. Bombrun and C. Germain are with Laboratoire IMS, Universit de Bordeaux, (e-mail: firstname.lastname@u-bordeaux.fr)

I. Ilea, R. Terebes and M. Borda are with Technical University of Cluj-Napoca, (e-mail: firstname.lastname@com.utcluj.ro)

following, we briefly recall the definition of the center of mass (CM) [20]–[22] and the median (Med) [2], [15] and next we introduce the proposed M-estimator.

a) The center of mass is one of the most popular estimators, for which the cost function is:

$$f_{CM}(\mathrm{M}) = \frac{1}{N}\sum_{i=1}^{N} d^2(\mathrm{M}, \mathrm{M}_i), \tag{2}$$

where $d(\cdot)$ represents the Rao's Riemannian distance between two covariance matrices defined as [23]:

$$d(\mathrm{M_1}, \mathrm{M_2}) = \left[\frac{1}{2}\sum_{i=1}^{m}(\ln \lambda_i)^2\right]^{\frac{1}{2}}, \tag{3}$$

where λ_i, $i = 1 \dots m$ are the eigenvalues of $\mathrm{M}_2^{-1}\mathrm{M}_1$.

Even though this method is largely used, it has a major drawback: it is easily influenced by the outliers present in the dataset [14], [15]. In order to reduce the impact of aberrant data on the estimated centroid's value, several possibilities are available. Some authors have proposed in [15], [16] the use of some trimming based methods to remove the outliers before the computation of (2). By deleting the elements that differ from the rest of the dataset, some new ones will become oultiers. If the removal procedure is repeated, the dataset may become too small for further reliable analysis. Therefore, a more appropriate solution is the use of robust methods for computing the centroid, like the median [15].

b) The median is defined by using the distance function:

$$f_{Med}(\mathrm{M}) = \frac{1}{N}\sum_{i=1}^{N} d(\mathrm{M}, \mathrm{M}_i). \tag{4}$$

It has to be mentioned that the estimation of the center of mass and the median from a set of covariance matrices have been recently studied in [3], [14], [15].

The center of mass and the median are two extreme solutions: the first one is efficient for datasets with no outliers, while the second one is robust to the presence of aberrant observations. In the following, we propose a trade-off between these two methods by introducing a Huber-like estimator.

B. The Huber's estimator

1) Definition of the Huber's centroid

In this section, we introduce a novel centroid estimator on the manifold of covariance matrices, based on the theory of M-estimators [17], [18], [24]. In this case, the cost function in (1), denoted $f_u(\mathrm{M})$ for the M-estimator, can be expressed by means of a scalar weight function $u(\cdot)$, as follows:

$$f_u(\mathrm{M}) = \frac{1}{N}\sum_{i=1}^{N} u\big(d(\mathrm{M}, \mathrm{M}_i)\big) d^2(\mathrm{M}, \mathrm{M}_i), \tag{5}$$

where $u(\cdot)$ is a positive-valued function which gives a weight to each observation M_i in the computation of the centroid. Obviously, the weight function $u(\cdot)$ should decrease to zero to ensure that the outliers have a smaller contribution to the centroid's estimate than the other observations. Note that even if the center of mass (2) and the median (4) have expressions similar to (5) for respectively $u(d(\mathrm{M}, \mathrm{M}_i)) = 1$ and $u(d(\mathrm{M}, \mathrm{M}_i)) = \frac{1}{d(\mathrm{M}, \mathrm{M}_i)}$, they do not belong to the family of M-estimators since the regularity conditions of their corresponding weight function $u(\cdot)$ defined in [24] are not satisfied.

In [17], Huber introduces the so-called Huber's function $u(\cdot)$ defined as:

$$u\big(d(\mathrm{M}, \mathrm{M}_i)\big) = \min\left(1, \frac{T}{d(\mathrm{M}, \mathrm{M}_i)}\right) \tag{6}$$

where T is a threshold value controlling the contribution of outliers in the estimation. By combining (5) and (6), the proposed Huber's centroid estimator is the covariance matrix M, which minimizes the following cost function:

$$\begin{aligned} f_H(\mathrm{M}) = \frac{1}{N}\sum_{i=1}^{N} d^2(\mathrm{M}, \mathrm{M}_i)\ \mathbb{1}_{\{d(\mathrm{M}, \mathrm{M}_i) \le T\}} + \\ + \frac{T}{N}\sum_{i=1}^{N} d(\mathrm{M}, \mathrm{M}_i)\ \mathbb{1}_{\{d(\mathrm{M}, \mathrm{M}_i\} > T\}}, \end{aligned} \tag{7}$$

where $\mathbb{1}_{\{a \le b\}}$ is the indicator function, which equals 1 if $a \le b$ and 0 otherwise. Threshold T represents a measure for discriminating between normal and aberrant data and therefore, it controls the estimator's behavior. In other words, for a large value of T, the Huber's estimator behaves as the center of mass, while for a small value it is equivalent to the median.

In this paper, we propose an algorithm to estimate the Huber's centroid by means of a gradient descent algorithm which minimizes the distance function given in (7). The gradient of $f_H(\mathrm{M})$ with respect to M that is $\nabla(f_H(\mathrm{M}))$ can be written as:

$$\begin{aligned} \nabla(f_H(\mathrm{M})) = -\frac{2}{N}\sum_{i=1}^{N} \mathrm{Log}_{\mathbf{M}}(\mathrm{M}_i)\ \mathbb{1}_{\{d(\mathrm{M}, \mathrm{M}_i) \le T\}} \\ -\frac{T}{N}\sum_{i=1}^{N} \frac{\mathrm{Log}_{\mathbf{M}}(\mathrm{M}_i)}{d(\mathrm{M}, \mathrm{M}_i)}\ \mathbb{1}_{\{d(\mathrm{M}, \mathrm{M}_i\} > T\}}, \end{aligned} \tag{8}$$

where $\mathrm{Log}_{\mathbf{M}}$ is the Riemannian logarithm mapping [25], [26]. Once that this value is obtained, the centroid can be updated as:

$$\mathrm{M}_{it+1} = \mathrm{Exp}_{\mathrm{M}_{it}}(-s_{it}\ \nabla(f_H(\mathrm{M}_{it}))), \tag{9}$$

with s_{it} being the descent step and $\mathrm{Exp}_{\mathbf{M}}$ the Riemannian exponential mapping [25], [26]. In practice, the Armijo's backtracking procedure [27] is used to fix s_{it} at each iteration of the algorithm.

This recursive process is repeated as long as the norm of $\nabla(f_H(\mathrm{M}_{it}))$, denoted D_{it}, is greater than a precision parameter ϵ, or until a maximum number of iterations $N_{\max}$ is reached. Practically, D_{it} is given as:

$$D_{it} = ||\nabla(f_H(\mathrm{M}_{it}))|| = \mathrm{tr}\left((\mathrm{M}_{it}^{-1}\nabla(f_H(\mathrm{M}_{it})))^2\right). \tag{10}$$

In the end, the Huber's centroid $\widehat{\mathrm{M}}_H$ estimator is obtained. A pseudo-code description of the Huber's centroid estimation is given in Algorithm 1.

As observed in (8), the first and the second terms correspond respectively to the gradient of the cost function for the center

Algorithm 1 Huber's centroid estimator

```
Input: $\mathrm{M}_1, \dots, \mathrm{M}_N$, $T$, $\epsilon$, $N_{\max}$
Initialize $\mathrm{M}$ using the sample mean
$it = 1$
while ($D_{it} > \epsilon$) and ($it \leq N_{\max}$) do
    Estimate $\mathrm{M}$ using one iteration of (9).
    Compute the gradient norm, $D_{it}$, according to (10).
    $it = it + 1$
end while
Output: $\mathrm{M}$
```

of mass and median centroids. For the second term, it can be seen that the division by distance $d(\mathrm{M}_{it}, \mathrm{M}_i)$ is needed. In some cases, that is when an observation M_i is close to the current centroid's estimate M_{it}, their distance is close to 0 yielding to potential numerical unsuitability. To avoid this, in [14] the author proposes to exclude, at each iteration it, the observations M_i that are too close from M_{it}. By using the proposed Huber's centroid, this problem is solved automatically by considering the threshold T. In conclusion, by choosing an appropriate value for T, the division by zero in the gradient (8) will be avoided, which represents an important advantage of the proposed method.

2) Determination of an automatic Huber's threshold

As explained before, the performance of the Huber's centroid estimator depends greatly on the threshold T that discriminates between aberrant and normal data. There is hence a need to fix it automatically or at least to give an idea of the order of magnitude of T. In practice, T is application dependent and is related to the intrinsic variability of the observed data. By considering first and second order statistics, the Riemannian Gaussian distribution (RGD) has been introduced in [26]. This distribution is characterized by two parameters: the central value $\bar{\mathrm{M}}$ and the dispersion σ. Its probability density function of the RGD is given by

$$p(\mathrm{M}|\bar{\mathrm{M}}, \sigma) = \frac{1}{Z(\sigma)} \exp\Big\{ - \frac{d^2(\mathrm{M}, \bar{\mathrm{M}})}{2\sigma^2} \Big\}, \tag{11}$$

where $Z(\sigma)$ is a normalization factor independent of the centroid $\bar{\mathrm{M}}$, and $d(\mathrm{M}, \bar{\mathrm{M}})$ is the Riemannian distance defined in (3).

In order to estimate the threshold's value, a robust estimator of the dispersion parameter σ is required. Inspired by previous works on robust statistics [28], we propose to extend the concept of median absolute deviation (MAD) to the case of covariance matrices which live in a Riemannian space. The MAD of the set $\mathrm{M}_1, \ \dots, \mathrm{M}_N$ is defined as the median of the Riemannian distances d computed between each sample M_i and the Riemannian median (denoted $RMed(\mathrm{M})$):

$$MAD = median(d(\mathrm{M}_i, RMed(\mathrm{M})). \tag{12}$$

Once the MAD is computed, the robust estimate $\hat{\sigma}$ of the RGD's dispersion can be obtained as:

$$\hat{\sigma} = \frac{K}{m} \times MAD, \tag{13}$$

where m is the size of covariance matrices and K is a constant depending on the distribution of $d\big(\mathrm{M}_i, RMed(\mathrm{M})\big)/\sigma$. More precisely, K is obtained by studying the statistics of $z = \frac{d(\mathrm{M},\bar{\mathrm{M}})}{m\sigma}$ since by definition of the MAD, we have:

$$\frac{1}{2} = p\big(d(\mathrm{M}, \bar{\mathrm{M}}) \leq MAD\big) = p\bigg(\frac{d(\mathrm{M}, \bar{\mathrm{M}})}{m\sigma} \leq \frac{MAD}{m\sigma}\bigg). \tag{14}$$

In practice, it has been observed on simulated data that the distribution of z is independent of $\bar{\mathrm{M}}$ and σ[1]. By combining (13) and (14), the constant $K = 1/\big(\phi^{-1}(0.5)\big)$, knowing that ϕ^{-1} is the inverse of the cumulative distribution function of z. Experiments have shown that $K \approx 1.312$. Finally, the Huber's threshold is obtained by multiplying the estimated standard deviation $\hat{\sigma}$ by a constant c, which will give $T = c \times \hat{\sigma}$. A common value for c is 1.5 as recommended in [28].

C. Performance Analysis

In the following, the influence of outliers on the proposed Huber's centroid estimator is studied. The obtained results are presented in this section and they are compared to those given by the center of mass and the median.

For this experiment, covariance matrices are generated as realizations of RGDs. For more information concerning the generation of samples from an RGD, the interested reader is referred to section III-A of [26]. In our case, the simulated covariance matrix datasets are obtained for centroids $\bar{\mathrm{M}}$ of size $m \times m$ having the form $\bar{\mathrm{M}}(i, j) = \rho^{|i-j|}$ for $i, j \in [\![1, m]\!]$.

Since the centroid is a covariance matrix, the manifold of the space of covariance matrices should be taken into account for the estimators' performance evaluation. In the literature, many authors have proposed to define the concept of intrinsic analysis for statistical estimation [29]–[31]. To this aim, the notions of intrinsic root-mean square error (RMSE) and intrinsic bias vector field have been introduced. We briefly recall here their definitions.

Let $\widehat{\bar{\mathrm{M}}}$ be the estimated centroid of the dataset, that is the estimate of the centroid $\bar{\mathrm{M}}$. The intrinsic RMSE is given by [29]–[31]:

$$RMSE = \sqrt{E\big[d^2(\widehat{\bar{\mathrm{M}}}, \bar{\mathrm{M}})\big]}, \tag{15}$$

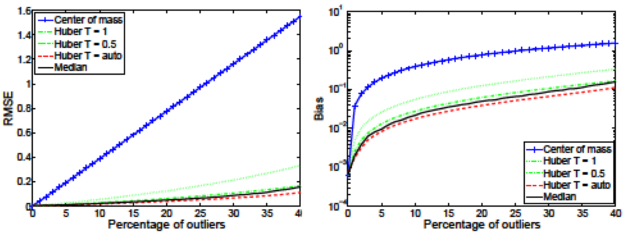
where $d(\cdot)$ is the Riemannian distance defined in (3). In addition, the bias vector field $b(\bar{\mathrm{M}})$ of $\widehat{\bar{\mathrm{M}}}$ is given by [29]–[31]:

$$\mathrm{b}(\bar{\mathrm{M}}) = \mathrm{Log}_{\bar{\mathrm{M}}} E_{\bar{\mathrm{M}}}\big[\widehat{\bar{\mathrm{M}}}\big] = E\big[\mathrm{Log}_{\bar{\mathrm{M}}} \widehat{\bar{\mathrm{M}}}\big], \tag{16}$$

knowing that $E_{\bar{\mathrm{M}}}\big[\widehat{\bar{\mathrm{M}}}\big] = \mathrm{Exp}_{\bar{\mathrm{M}}} E\big[\mathrm{Log}_{\bar{\mathrm{M}}} \widehat{\bar{\mathrm{M}}}\big]$. Since the bias vector field $\mathrm{b}(\bar{\mathrm{M}})$ in (16) is a covariance matrix, we compute its norm according to (10) to plot it in the following figures.

To study the influence of outliers on the centroid's estimation, a dataset containing 1000 matrices of size 2×2 is created. These matrices have an RGD distribution of dispersion $\sigma = 0.1$ and centroid $\bar{\mathrm{M}}$ obtained for $\rho = 0.7$. To this original data set, some outliers are added. They are i.i.d. covariance matrices samples issued from an RGD of centroid $10 \times M_o$, with M_o obtained for $\rho_o = 0.1$. Here, the dispersion for the outlier samples σ_o is set to 0.1.

[1] The use of z is equivalent to the standardization step $z = \frac{x-\mu}{\sigma}$ for a univariate normal distribution.



(a) (b)

Fig. 1. The RMSE (a) and the bias vector field (b) as functions of the outlier percentage.

Figs. 1 draws the results obtained for the intrinsic RMSE (a) and for the intrinsic bias vector field (b) as functions of the percentage of outliers, knowing that 5000 Monte Carlo runs have been used to evaluate the estimators' performance. The behavior of the center of mass (in blue), the median (in black) and the Huber's centroid with fixed threshold $T = 1$ and $T = 0.5$ (in green) and automatically computed value for T (in red) are analyzed, when the percentage of aberrant data varies from 0 to 40%. As observed, the center of mass is clearly influenced by the presence of outliers while for robust estimators, like the median or the Huber's centroid, this influence in less important.

III. Application to MEG Based Brain Decoding

In this section, we apply the proposed centroid estimator to brain decoding, based on MEG data. The database used for the Biomag 2014 Decoding Challenge: Brain Decoding Across Subjects (DecMeg2014) [32] has been considered. The idea of brain decoding consists in predicting the stimulus presented to the subject from the concurrent brain activity [33]. For this experiment, two categories of visual stimulus have been considered: face and scrambled face. Therefore, the problem to solve can be viewed as a two-class classification task. A detailed description of the neuroscientific experiment implemented to collect the data can be found in [34].

The database contains 16 training and 7 testing subjects. For each training subject, approximately 580 trials have been considered, giving a training set of 9414 trials. Next, for each trial, covariance matrices of size 16×16 have been extracted, as described in [35]. Further on, a modified version of the unsupervised classification method presented in [35] has been implemented. First, a regularized logistic regression model has been trained to obtain the initial labels for the unsupervised classification algorithm (k-means). Second, the centroids of each class (face or scrambled face) are computed. For this step, several estimators have been considered: the center of mass, the median, the Huber's estimator with both fixed ($T = 0.2$ and $T = 0.5$) and automatically computed thresholds and also the trimmed based methods [16], when $d = 5\%$ of discarded extreme data. For this latter, only the best result has been retained, that is the mean-based trimmed median. Next, for each testing subject, covariance matrices have been computed and the classification has been performed by two approaches. First, the winner method of the DecMeg2014 competition has been implemented, for which the test trials have been assigned to the closest class, by using the minimum distance to mean (MDM) Riemannian classifier [36]. Second, the covariance matrices have been modeled as RGDs and each trial has been assigned to the centroid maximizing the log-likelihood criterion derived from (11).

TABLE I
Classification results for MEG based brain decoding.

Estimator	MDM	MGD
CM	74.106	73.845
Med	73.627	74.150
Huber $T = 0.2$	**74.847**	**75.109**
Huber $T = 0.5$	74.063	73.976
Huber $T = auto$	74.455	74.106
Trimming ($d = 5\%$) [16]	74.412	74.542

The obtained results are shown in Table I and several remarks can be made. By analyzing the above table, it can be seen that the use of Huber's estimator may increase the classification performance. The obtained values are comparable or higher to those given by the other robust estimators, but without their disadvantages: division by zero for the median, or choice of the percentage d of discarded observation for the trimmed estimators. Interestingly, note that the estimated Hubers's threshold T is recomputed at each k-means iteration. And in this experiment, it varies between 0.38 and 0.46 across the test subjects and the classes. Moreover, the proposed estimated value of T by the MAD gives an order of magnitude of the threshold we may consider in the Huber estimation algorithm. This value can be readjusted to improve the classification performance as observed in Table I.

IV. Conclusion

In this article, a new method called the Huber's centroid, for the estimation of the central value of a covariance matrix dataset has been introduced. This estimator is a trade-off between the center of mass and the median. The definition of the Huber's centroid and its computational algorithm have been detailed. In addition, an algorithm for choosing the appropriate threshold value for the Huber's estimator has been developed. Further on, the Huber's centroid, has been applied to the case of covariance matrices representing realizations of Riemannian Gaussian distributions. The robustness to outlier values has been studied on simulated data, but also in the context of brain decoding, that is a two-class classification experiment. The results have been compared to those given by two well-known estimators that are the center of mass and the median but also to those given by trimmed based methods.

Further works will include the statistical modeling of $z = d(\mathrm{M}_i, \bar{\mathrm{M}})/m\sigma$ to derive the analytical expression of K. In addition, the proposed centroid will be used to build the codebook for patch-based image classification algorithms.

Acknowledgment

This study has been carried out in the frame of the Investments for the future Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02) of the French National Research Agency (ANR). It has also been supported by the French Foreign Affairs and International Development Ministry and by the Executive Agency for Higher Education, Research, Development and Innovation Funding Romania, under the projects 32619VL and PNII Capacitati 779/27.06.2014.

References

[1] E. Ollila and V. Koivunen, "Robust antenna array processing using M-estimators of pseudo-covariance," in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, vol. 3, Sept 2003, pp. 2659–2663.

[2] L. Yang, M. Arnaudon, and F. Barbaresco, "Riemannian median, geometry of covariance matrices and radar target detection," *European Radar Conference (EuRAD)*, pp. 415–418, 2010.

[3] F. Barbaresco, M. Arnaudon, and L. Yang, "Riemannian medians and means with applications to radar signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 4, pp. 595–604, 2013.

[4] R. de Luis-García, C.-F. Westin, and C. Alberola-López, "Gaussian mixtures on tensor fields for segmentation: Applications to medical imaging," *Computerized Medical Imaging and Graphics*, vol. 35, no. 1, pp. 16–30, 01 2011.

[5] J. Robinson, "Covariance matrix estimation for appearance-based face image processing," *Proceedings of the British Machine Vision Conference 2005*, pp. 389–398, 2005. [Online]. Available: http://www.intuac.com/userport/john/pubs/covestbmvc.pdf

[6] K. Mader and G. Reese, "Using covariance matrices as feature descriptors for vehicle detection from a fixed camera," *ArXiv e-prints*, Feb. 2012.

[7] P. Formont, F. Pascal, G. Vasile, J. Ovarlez, and L. Ferro-Famil, "Statistical classification for heterogeneous polarimetric SAR images," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 567–576, June 2011.

[8] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *NeuroComputing*, vol. 112, pp. 172–178, 2013.

[9] S. Said, L. Bombrun, and Y. Berthoumieu, "Texture classification using Rao's distance on the space of covariance matrices," in *Geometric Science of Information (GSI)*, 2015.

[10] M. Faraki, M. Harandi, and F. Porikli, "More about VLAD: A leap from Euclidean to Riemannian manifolds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*, June 2015, pp. 4951–4960.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, Oct. 2007.

[12] B. Afsari, "Riemannian lp center of mass: existence, uniqueness and convexity," *Proceedings of the American Mathematical Society*, vol. 139, no. 2, pp. 655–673, 2011.

[13] P. Formont, J.-P. Ovarlez, and F. Pascal, "On the use of matrix information geometry for polarimetric SAR image classification," in *Matrix Information Geometry*, F. Nielsen and R. Bhatia, Eds. Springer Berlin Heidelberg, 2013, pp. 257–276.

[14] L. Yang, "Riemannian median and its estimation," *LMS Journal of Computation and Mathematics*, vol. 13, pp. 461–479, 2010.

[15] P. T. Fletcher, S. Venkatasubramanian, and S. C. Joshi, "Robust statistics on Riemannian manifolds via the geometric median," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2008.4587747

[16] T. Uehara, T. Tanaka, and S. Fiori, *Advances in Cognitive Neurodynamics (V): Proceedings of the Fifth International Conference on Cognitive Neurodynamics - 2015*. Singapore: Springer Singapore, 2016, ch. Robust averaging of covariance matrices by Riemannian geometry for motor-imagery brain–computer interfacing, pp. 347–353.

[17] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[18] D. E. Tyler, "A distribution-free M-estimator of multivariate scatter," *The Annals of Statistics*, vol. 15, no. 1, pp. 234–251, 03 1987. [Online]. Available: http://dx.doi.org/10.1214/aos/1176350263

[19] X. Gu, J. Deng, and M. Purvis, "Improving superpixel-based image segmentation by incorporating color covariance matrix manifolds," in *IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 4403–4406.

[20] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on Pure and Applied Mathematics*, vol. 30, no. 5, pp. 509–541, 1977. [Online]. Available: http://dx.doi.org/10.1002/cpa.3160300502

[21] F. Nielsen and R. Bhatia, *Matrix Information Geometry*. Springer Berlin Heidelberg, 2012.

[22] S. Fiori, "Learning the Fréchet mean over the manifold of symmetric positive-definite matrices," *Cognitive Computation*, vol. 1, no. 4, pp. 279–291, 2009.

[23] G. Verdoolaege and P. Scheunders, "On the geometry of multivariate generalized Gaussian models," *Journal of Mathematical Imaging and Vision*, vol. 43, no. 3, pp. 180–193, 2012.

[24] R. A. Maronna, "Robust M-estimators of multivariate location and scatter," *Annals of Statistics*, vol. 4, no. 1, pp. 51–67, Jan. 1976.

[25] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006. [Online]. Available: http://dx.doi.org/10.1007/s11263-005-3222-z

[26] S. Said, L. Bombrun, and Y. Berthoumieu, "Riemannian Gaussian distributions on the space of symmetric positive definite matrices," *available on arxiv via http://arxiv.org/abs/1507.01760*, 2015.

[27] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives." *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966. [Online]. Available: http://projecteuclid.org/euclid.pjm/1102995080

[28] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. John Wiley & Sons, Inc., 2009.

[29] J. M. Oller and J. M. Corcuera, "Intrinsic analysis of statistical estimation," *The Annals of Statistics*, vol. 23, no. 5, pp. 1562–1581, 1995. [Online]. Available: http://dx.doi.org/10.2307/2242534

[30] S. Smith, "Covariance, subspace, and intrinsic Cramér-Rao bounds," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1610–1630, May 2005.

[31] G. Garcia and J. M. Oller, "What does intrinsic mean in statistical estimation?" *Statistics and Operations Research Transactions*, vol. 30, no. 2, pp. 125–170, 2006.

[32] "DecMeg2014 - Decoding the Human Brain," Biomag 2014 Decoding Challenge: Brain Decoding Across Subjects. Available: https://www.kaggle.com/c/decoding-the-human-brain.

[33] E. Olivetti, S. M. Kia, and P. Avesani, "MEG decoding across subjects," in *International Workshop on Pattern Recognition in Neuroimaging, PRNI, 2014, Tübingen, Germany, June 4-6, 2014*, 2014, pp. 1–4.

[34] R. N. Henson, D. G. Wakeman, V. Litvak, and K. J. Friston, "A parametric empirical bayesian framework for the EEG/MEG inverse problem: generative models for multisubject and multimodal integration," *Frontiers in Human Neuroscience*, vol. 5, no. 76, 2011.

[35] A. Barachant, "MEG decoding using Riemannian geometry and unsupervised classification," *Technical Report*. [Online]. Available: http://alexandre.barachant.org/wp-content/uploads/2014/08/documentation.pdf

[36] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain-computer interface classification by riemannian geometry." *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, 2012.

Texture Image Classification with Riemannian Fisher Vectors

I. Ilea, L. Bombrun, C. Germain, R. Terebes, M. Borda and Y. Berthoumieu. *Texture image classification with Riemannian Fisher vectors.* In IEEE International Conference on Image Processing, pages 3543 – 3547, 2016

TEXTURE IMAGE CLASSIFICATION WITH RIEMANNIAN FISHER VECTORS

Ioana Ilea[1,2]*, Lionel Bombrun*[1]*, Christian Germain*[1]*, Romulus Terebes*[2]*, Monica Borda*[2] *and Yannick Berthoumieu*[1]

[1]: Université de Bordeaux, Laboratoire IMS, Groupe Signal et Image.
{*ioana.ilea, lionel.bombrun, christian.germain, yannick.berthoumieu* }*@ims-bordeaux.fr*
[2]: Technical University of Cluj-Napoca. {*romulus.terebes, monica.borda* }*@com.utcluj.ro*

ABSTRACT

This paper introduces a generalization of the Fisher vectors to the Riemannian manifold. The proposed descriptors, called Riemannian Fisher vectors, are defined first, based on the mixture model of Riemannian Gaussian distributions. Next, their expressions are derived and they are applied in the context of texture image classification. The results are compared to those given by the recently proposed algorithms, bag of Riemannian words and R-VLAD. In addition, the most discriminant Riemannian Fisher vectors are identified.

***Index Terms*—** Riemannian Fisher vectors, bag of words, Riemannian Gaussian distributions, classification, covariance matrix.

1. INTRODUCTION

Bag of words, Fisher vectors, or vectors of locally aggregated descriptors represent some of the most frequently used local models in order to capture the information lying in signals [1], images [2] or videos [3]. These descriptors have multiple advantages. First, the obtained information can be used in a wide variety of applications like classification [2] and categorization [4], text [5] and image [6] retrieval, action and face recognition [7], etc. Second, combined with powerful local feature descriptors such as SIFT, they are robust to transformations like scaling, translation, or occlusion [7].

The *bag of words* (BoW) model has been used for text retrieval and categorization [5, 8] and then extended to visual categorization [9]. This method is based on the construction of a codebook, or a dictionary, that contains the most significant features in the dataset. Generally, the elements in the codebook, or the words, are the clusters' centroids obtained by using the conventional k-means clustering algorithm. Next, for each element in the dataset, its signature is determined by computing the histogram of the number of occurrences of each word in its structure. To improve the performance of BoW, which counts only the number of local descriptors assigned to each Voronoi region, *Fisher vectors* (FV) have been introduced by including other statistics, such as the mean and variance of local descriptors.

FV are descriptors based on Fisher kernels [1], representing methods for measuring if samples are correctly fitted by some given models. By using FV, a sample is characterized by the gradient vector of the probability density function that models it, classically a Gaussian mixture model (GMM) [4]. In practice, the probability density function is replaced by the log-likelihood and, as mentioned in [4], its gradient describes the direction in which parameters should be modified to best fit the data. The derivatives with respect to the model's parameters are computed and concatenated to obtain the FV.

The *vectors of locally aggregated descriptors* (VLAD) represent a simplification of the Fisher kernel [10], based on the definition of a codebook. In the computation process, first of all, the dictionary has to be built. For this reason, the dataset is partitioned by using a clustering algorithm and the cluster centroids represent the codebook elements. Next, each element in the dataset is associated to the closest cluster. Further on, for each cluster a vector is computed, containing the differences between the cluster's centroid and each element in that cluster. In the end, the sum of differences concerning each cluster is computed and the final VLAD feature vector is given by the concatenation of all the previously obtained sums. In other way, the VLAD descriptors can be obtained starting from FV, by taking into consideration only the derivatives with respect to the means of the GMM. Note also that the homoscedasticity assumption and the hard assignment scheme are required to obtain VLAD features [7, 10].

Those three approaches have been widely used for many applications involving non-parametric features. Recently BoW and VLAD have been extended to the case where each feature is a point on a Riemannian manifold. This is for instance the case where local descriptors are covariance matrices. This includes many different applications in image processing, like classification [11, 12, 13], image segmentation [14], object detection [15, 16], etc. In [3] and [17], the BoW approach has been extended to the so-called log-Euclidean bag of words (LE-BoW) and bag of Riemannian words (BoRW) models by considering respectively the log-Euclidean and geodesic distance between two points on the manifold. In addition, the Riemannian version of the VLAD method (R-VLAD) has been developed in [7] and has shown superior classification performances, compared to the classic VLAD algorithm.

Until now, FV have not yet been generalized in the same manner to Riemannian manifold, due to the lack of probabilistic generative models suited for parametric descriptors. This represents the main contribution of this paper. The proposed Riemannian Fisher vectors (RFV) are a generalization of the FV for parametric descriptors based on the recent works on the definition of the Riemannian Gaussian distributions (RGDs) [18].

The paper is structured as follows. Section 2 recalls some elements on the RGD like its definition, the expression of mixtures of RGDs and the parameter's estimation procedure. Section 3 introduces the definition of the proposed RFV, their computation and their relation with R-VLAD. Section 4 presents an application of the proposed RFV to texture image classification. Conclusions and future works are finally reported in Section 5.

2. RIEMANNIAN GAUSSIAN DISTRIBUTIONS

Let $\mathbf{\Upsilon} = \{\mathbf{Y}_t\}_{t=1:T}$ be a set of T independent and identically distributed (i.i.d.) samples according to a Riemannian Gaussian distribution of central value $\bar{\mathbf{Y}}$ and dispersion σ. The probability density function of the RGD with respect to the Riemannian volume element, in the space $\mathcal{P}_m$ of $m \times m$ real, symmetric and positive definite matrices, has been introduced in [18] as:

$$p(\mathbf{Y}_t|\bar{\mathbf{Y}}, \sigma) = \frac{1}{Z(\sigma)} \exp\Big\{ - \frac{d^2(\mathbf{Y}_t, \bar{\mathbf{Y}})}{2\sigma^2} \Big\}, \qquad (1)$$

where $Z(\sigma)$ is a normalization factor independent of the centroid $\bar{\mathbf{Y}}$ and $d(\cdot)$ is the Riemannian distance given by $d(\mathbf{Y}_1, \mathbf{Y}_2) = \left[\sum_i (\ln \lambda_i)^2\right]^{\frac{1}{2}}$, with λ_i, $i = 1, \dots, m$ being the eigenvalues of $\mathbf{Y}_2^{-1}\mathbf{Y}_1$.

Starting from (1), the probability density function for a mixture of K RGDs can be defined as [18]:

$$p(\mathbf{Y}_t|\lambda) = \sum_{j=1}^{K} \varpi_j \, p(\mathbf{Y}_t|\bar{\mathbf{Y}}_j, \sigma_j), \qquad (2)$$

where $\lambda = \{(\varpi_j, \bar{\mathbf{Y}}_j, \sigma_j)_{1\leq j\leq K}\}$ is the parameter vector. ϖ_j are positive weights, with $\sum_{j=1}^{K} \varpi_j = 1$ and $p(\mathbf{Y}_t|\bar{\mathbf{Y}}_j, \sigma_j)$ is given by (1).

Several approaches can be employed to estimate the parameters $\{\widehat{\bar{\mathbf{Y}}}_j, \hat{\sigma}_j, \hat{\varpi}_j\}_{1\leq j\leq K}$ of the mixture of K RGDs [12]. The simplest one implies the estimation of the centroids $\widehat{\bar{\mathbf{Y}}}_j$, of clusters c_j, $j = 1, \dots, K$ by using the intrinsic k-means algorithm on a Riemannian manifold [7]. Thus, for each cluster c_j, the cost function

$$\varepsilon(\bar{\mathbf{Y}}_j) = \frac{1}{N_j} \sum_{n=1}^{N_j} d^2(\bar{\mathbf{Y}}_j, \mathbf{Y}_{j_n}) \qquad (3)$$

has to be minimized, where $\mathbf{Y}_{j_n}$ is the set of elements $\mathbf{Y}_j$ in cluster c_j, $n = 1, \dots, N_j$ and N_j is the cardinal of $\mathbf{Y}_{j_n}$. The minimizer of the cost function defined in (3) is known to be the Riemannian centre of mass of this set. The interested reader is referred to [19] and [20] for an algorithm to compute the empirical Riemannian centre of mass. Next, for each cluster c_j, the estimated dispersion parameter $\hat{\sigma}_j$ is obtained as the solution of:

$$\sigma_j^3 \times \frac{d}{d\sigma_j} Z(\sigma_j) = \varepsilon(\widehat{\bar{\mathbf{Y}}}_j). \qquad (4)$$

This latter is solved by a conventional Newton-Raphson algorithm [12]. Finally, the estimated weights $\hat{\varpi}_j$ are given by:

$$\hat{\varpi}_j = \frac{N_j}{\sum_{j=1}^{K} N_j}. \qquad (5)$$

All the elements recalled in this part are applied in the next section to the definition of the proposed Riemannian Fisher vectors.

3. RIEMANNIAN FISHER VECTORS

3.1. Definition

Let $\mathbf{\Upsilon} = \{\mathbf{Y}_t\}_{t=1:T}$ be a sample of T i.i.d observations following a mixture of K RGDs. Under the independence assumption, the probability density function of $\mathbf{\Upsilon}$ is given by:

$$p(\mathbf{\Upsilon}|\lambda) = \prod_{t=1}^{T} p(\mathbf{Y}_t|\lambda), \qquad (6)$$

where $\lambda = \{(\varpi_j, \bar{\mathbf{Y}}_j, \sigma_j)_{1\leq j\leq K}\}$ is the parameter vector and $p(\mathbf{Y}_t|\lambda)$ is the probability density function given in (2).

By using the Fisher kernels, the sample is characterized by its deviation from the model [2]. This deviation is measured by computing the Fisher score $U_{\mathbf{\Upsilon}}$ [1], that is the gradient ∇ of the log-likelihood with respect to the model parameters λ:

$$U_{\mathbf{\Upsilon}} = \nabla_\lambda \log p(\mathbf{\Upsilon}|\lambda) = \nabla_\lambda \sum_{t=1}^{T} \log p(\mathbf{Y}_t|\lambda). \qquad (7)$$

As mentioned in [1], the gradient of the log-likelihood with respect to a parameter describes the contribution of that parameter to the generation of a particular observation. In practice, a large value for this derivative is equivalent to a large deviation from the model. Further on, that can be translated into the fact that the model does not correctly fit the data.

In the following, the derivatives for the mixture of RGDs, are given, knowing that $\gamma_i(\mathbf{Y}_t)$ is the probability that the observation $\mathbf{Y}_t$ is generated by the i^{th} RGD and it is computed as:

$$\gamma_i(\mathbf{Y}_t) = \frac{\varpi_i \, p(\mathbf{Y}_t|\bar{\mathbf{Y}}_i, \sigma_i)}{\sum_{j=1}^{K} \varpi_j \, p(\mathbf{Y}_t|\bar{\mathbf{Y}}_j, \sigma_j)}. \qquad (8)$$

To determine the gradient with respect to the weight, we consider the procedure described in [2]. For that, the following parametrization is used in order to ensure the positivity and sum to one constraints of the weights:

$$\varpi_i = \frac{\exp(\alpha_i)}{\sum_{j=1}^{K} \exp(\alpha_j)}. \tag{9}$$

By taking into consideration all these observations, the derivatives with respect to the parameters in λ can be obtained as:

$$\frac{\partial \log p(\Upsilon|\lambda)}{\partial \bar{\mathbf{Y}}_i} = \sum_{t=1}^{T} \gamma_i(\mathbf{Y}_t)\, \sigma_i^{-2}\, \mathrm{Log}_{\bar{\mathbf{Y}}_i}(\mathbf{Y}_t), \tag{10}$$

$$\frac{\partial \log p(\Upsilon|\lambda)}{\partial \sigma_i} = \sum_{t=1}^{T} \gamma_i(\mathbf{Y}_t) \left\{ -\frac{Z'(\sigma_i)}{Z(\sigma_i)} + \frac{d^2(\mathbf{Y}_t, \bar{\mathbf{Y}}_i)}{\sigma_i^3} \right\}, \tag{11}$$

$$\frac{\partial \log p(\Upsilon|\lambda)}{\partial \alpha_i} = \sum_{t=1}^{T} \gamma_i(\mathbf{Y}_t)\, (1 - \varpi_i), \tag{12}$$

where $\mathrm{Log}_{\bar{\mathbf{Y}}_i}(\cdot)$ is the Riemannian logarithm mapping.

The vectorized representation of the derivatives in (10), (11) and (12) of the log-likelihood, with respect to the parameters in λ, gives the Riemannian Fisher vectors (RFV). In the end, by using the RFV, a sample is characterized by a feature vector containing some, or all the derivatives, having the maximum length given by the number of parameters in λ.

3.2. Relation with R-VLAD

As mentioned earlier in the introduction, VLAD features are a special case of FV. Therefore, R-VLAD can be viewed as a particular case of the proposed RFV. More precisely, R-VLAD is obtained by taking into consideration only the derivatives with respect to the central value $\bar{\mathbf{Y}}_i$ (see (10)). In addition, a hard assignment scheme is applied. Starting from the definition of the elements v_i in the R-VLAD descriptor [7]:

$$v_i = \sum_{\mathbf{Y}_t \in c_i} \mathrm{Log}_{\bar{\mathbf{Y}}_i}(\mathbf{Y}_t), \tag{13}$$

with $\mathbf{Y}_t \in c_i$ being the elements $\mathbf{Y}_t$ assigned to the cluster c_i, $i = 1, \dots, K$, the hard assignment implies that:

$$\gamma_i(\mathbf{Y}_t) = \begin{cases} 1, & \text{if } \mathbf{Y}_t \in c_i \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Moreover, the assumption of homoscedasticity is considered, that is $\sigma_i = \sigma$, $\forall i = 1, \dots, K$. By considering these two assumptions, it is clear that (10) reduces to (13) hence confirming that RFV are a generalization of R-VLAD descriptors.

4. APPLICATION TO TEXTURE IMAGE CLASSIFICATION

This section introduces an application to texture image classification. The aim of this experiment is first to analyze the potential of the proposed RFV compared to the recently proposed bag of Riemannian words (BoRW) model [17] and R-VLAD [7]. The BoRW, RFV and R-VLAD are built based on region covariance descriptors [21] containing basic information, like image intensity and gradients. The experiment's purpose is not to find the best classification rates, but to compare the two methods starting from very simple descriptors. Second, the objective is to determine the RFV that are the most discriminant to retrieve the classes: the one associated to $\bar{\mathbf{Y}}_i$ (10), σ_i (11) or α_i (12).

4.1. Databases

For this work, two texture databases are used: the VisTex [22] database and the Outex_TC000_13 [23] database. The *VisTex* database consists in 40 texture classes. Each class is composed of 64 images of size 64×64 pixels. The *Outex_TC000_13* database contains 68 texture classes, where each class is represented by a set of 20 images of size 128×128 pixels. For both databases, the feature extraction and classification steps are similar and are detailed in the next subsection.

4.2. Feature extraction and classification

For the classification procedure, the considered database is equally and randomly divided in order to obtain the training and the testing sets. For each image in the two sets, local descriptors have to be extracted first. In this experiment, the region covariance descriptors (RCovDs) are considered. In order to build the RCovD for an image I of size $W \times H$, several characteristics are extracted for each pixel $(x, y) \in I$. Here, the image intensities $I(x, y)$ and the norms of the first and second order derivatives of $I(x, y)$ in both directions x and y are considered [21]. Thus, a vector v of 5 elements is obtained for each pixel having the spatial position $(x, y) \in I$:

$$\mathbf{v}(x, y) = \left[I(x, y), \left| \frac{\partial I(x,y)}{\partial x} \right|, \left| \frac{\partial I(x,y)}{\partial y} \right|, \left| \frac{\partial^2 I(x,y)}{\partial x^2} \right|, \left| \frac{\partial^2 I(x,y)}{\partial y^2} \right| \right]^T. \tag{15}$$

For the two considered databases, the extracted RCovD are the estimated covariance matrices of vectors $\mathbf{v}(x, y)$ computed on a sliding patch of size 15×15 pixels. As an overlap of 8 pixels is considered for the patches, the VisTex and Outex databases are represented respectively by a set of 36 and 196 covariance matrices per texture class (of size 5×5). To speed-up the computation time, the fast covariance computation algorithm based on integral images presented in [21] has been implemented. In the end, each texture class is characterized by a set $\mathbf{Y}_1, \dots, \mathbf{Y}_N$ of N covariance matrices, that are elements in $\mathcal{P}_5$. Based on the patches in the training set, a codebook is created. For each class, the codewords are represented by the estimated parameters $\{\widehat{\bar{\mathbf{Y}}}_j, \hat{\sigma}_j, \hat{\varpi}_j\}_{1 \le j \le K}$ of the mixture of K RGDs defined in (2). The estimation procedure is carried out here by using the intrinsic k-means algorithm (see Section 2). For this experiment, the number of modes K is set to 3. In the end, the codebook is obtained by concatenating the previously extracted codewords.

Starting from the RCovDs and the learned codebook, the BoRW, RFV and R-VLAD local models are derived, as presented in the previous section. After their computation, a normalization stage is performed. In the RFV framework, the classical power and ℓ_2 normalizations are applied [17]. The ℓ_2 normalization has been proposed in [24] to minimize the influence of the background information on the image signature, while the power normalization corrects the independence assumption made on the patches [25]. The same normalization scheme is also applied for R-VLAD models. For the BoRW algorithm, only ℓ_2 normalization is performed, as recommended in [3].

For the classification step, the SVM algorithm with Gaussian kernel is considered, knowing that the dispersion parameter of the Gaussian kernel is optimized by using a cross validation procedure on the training set.

4.3. Results

The classification performances in term of overall accuracy on the VisTex and Outex_TC000_13 databases are reported in Tables 1 and 2 respectively. Those results are displayed for 10 random partitions in training and testing sets. Columns homoscedasticity and prior correspond respectively to the homoscedasticity assumption and to the use of the weights ϖ_i in the decision rule. If the homoscedasticity assumption is true, the dispersion parameter σ_i is the same for all the clusters c_i. If the prior parameter is set to false, all the clusters have the same weight. Note that for the BoRW approach published in [17] and the R-VLAD presented in [7], the dispersion and weight parameters were not considered. Note also that for the proposed RFV, those two parameters are respectively set to "false" and "true", since both the dispersion and weight parameters are considered in the derivation of the RFV.

In this experiment, we also analyze the contribution of each parameter (weight, dispersion and centroid) to the classification accuracy. For example, the row "RFV : ϖ" indicates the classification results when only the derivatives with respect to the weights are considered to calculate the RFV (see (12)), ...

As observed in Tables 1 and 2, the proposed RFV outperforms the BoRW and R-VLAD approaches. A gain of 1 to 3% is observed for the VisTex database. Moreover, among the RFVs types, the most discriminant feature is obtained by combining the derivatives with respect to all three parameters: centroid, dispersion and weight (see (10), (11), (12)).

5. CONCLUSION

In this paper, a new local model for image classification in the Riemannian space has been proposed. The introduced method, called Riemannian Fisher vectors, is a generalization of the so-called Fisher vectors, when the features are represented by parametric descriptors, like covariance matrices. The definition and the expression of RFV have been given, starting from the definition of the mixture of Riemannian Gaussian distributions. In addition, its relation with R-VLAD has been illustrated. In the end, the RFVs have been applied for texture image classification on the VisTex and Outex_TC000_13 databases. The results have been compared with those given by BoRW and R-VLAD, showing better classification rates for the same codebook. In addition, it has been observed that the most discriminant feature is obtained by combining the derivatives with respect to all parameters.

Method	Homoscedasticity	Prior	Overall accuracy
BoRW	false	true	87.22 ± 1.19
BoRW	false	false	87.51 ± 0.92
BoRW [17]	true	false	**87.20 ± 1.55**
BoRW	true	true	76.67 ± 2.35
RFV : ϖ	false	true	90.31± 0.94
RFV : σ	false	true	81.42 ± 1.12
RFV : $\bar{\mathbf{Y}}$	false	true	87.22 ± 1.15
RFV : σ, ϖ	false	true	83.05 ± 1.15
RFV : $\bar{\mathbf{Y}}, \varpi$	false	true	87.85 ± 0.97
RFV : $\bar{\mathbf{Y}}, \sigma$	false	true	90.41 ± 0.86
RFV : $\bar{\mathbf{Y}}, \sigma, \varpi$	false	true	**90.43 ± 0.84**
R-VLAD [7]	true	false	**87.94 ± 0.58**

Table 1. Classification results on the VisTex database.

Method	Homoscedasticity	Prior	Overall accuracy
BoRW	false	true	84.32 ± 0.99
BoRW	false	false	84.37± 1.28
BoRW [17]	true	false	**84.43 ± 1.23**
BoRW	true	true	79.31 ± 1.86
RFV : ϖ	false	true	84.94 ± 1.12
RFV : σ	false	true	78.46 ± 1.54
RFV : $\bar{\mathbf{Y}}$	false	true	83.94 ± 0.90
RFV : σ, ϖ	false	true	80.38 ± 1.80
RFV : $\bar{\mathbf{Y}}, \varpi$	false	true	84.26 ± 0.75
RFV : $\bar{\mathbf{Y}}, \sigma$	false	true	84.32 ± 1.19
RFV : $\bar{\mathbf{Y}}, \sigma, \varpi$	false	true	**84.12 ± 1.15**
R-VLAD [7]	true	false	**82.99 ± 1.19**

Table 2. Classification results on the Outex database.

Further works on this subject will concern the extension of RFV to the recently proposed mixture of Riemannian Laplace distributions [26, 27].

Acknowledgments

This study has been carried out in the frame of the Investments for the future Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02) of the French National Research Agency (ANR). It has also been supported by the French Foreign Affairs and International Development Ministry and by the Executive Agency for Higher Education, Research, Development and Innovation Funding Romania, under the projects 32619VL and PNII Capacitati 779/27.06.2014.

6. REFERENCES

[1] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *In Advances in Neural Information Processing Systems 11*. 1998, pp. 487–493, MIT Press.

[2] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[3] M. Faraki, M. Palhang, and C. Sanderson, "Log-euclidean bag of words for human action recognition," *Computer Vision, IET*, vol. 9, no. 3, pp. 331–339, 2015.

[4] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*. IEEE, 2007, pp. 1–8.

[5] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, Aug. 1988.

[6] M. Douze, A. Ramisa, and C. Schmid, "Combining attributes and Fisher vectors for efficient image retrieval," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2011, CVPR '11, pp. 745–752, IEEE Computer Society.

[7] M. Faraki, M.T. Harandi, and F. Porikli, "More about VLAD: A leap from Euclidean to Riemannian manifolds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*, June 2015, pp. 4951–4960.

[8] T. Joachims, *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings*, chapter Text categorization with Support Vector Machines: Learning with many relevant features, pp. 137–142, Springer Berlin Heidelberg, 1998.

[9] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[10] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision & Pattern Recognition*, jun 2010.

[11] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *NeuroComputing*, vol. 112, pp. 172–178, 2013.

[12] S. Said, L. Bombrun, and Y. Berthoumieu, "Texture classification using Rao's distance on the space of covariance matrices," in *Geometric Science of Information (GSI)*, 2015.

[13] I. Ilea, L. Bombrun, C. Germain, R. Terebes, and M. Borda, "Statistical hypothesis test for robust classification on the space of covariance matrices," in *IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 271–275.

[14] G. Garcia and J. M. Oller, "What does intrinsic mean in statistical estimation?," *Statistics and Operations Research Transactions*, vol. 30, no. 2, pp. 125–170, 2006.

[15] K. Mader and G. Reese, "Using covariance matrices as feature descriptors for vehicle detection from a fixed camera," *ArXiv e-prints*, Feb. 2012.

[16] J. Robinson, "Covariance matrix estimation for appearance-based face image processing," *Proceedings of the British Machine Vision Conference 2005*, pp. 389–398, 2005.

[17] M. Faraki, M. T. Harandi, A. Wiliem, and B. C. Lovell, "Fisher tensors for classifying human epithelial cells," *Pattern Recognition*, vol. 47, no. 7, pp. 2348 – 2359, 2014.

[18] S. Said, L. Bombrun, and Y. Berthoumieu, "Riemannian Gaussian distributions on the space of symmetric positive definite matrices," *available on arxiv via http://arxiv.org/abs/1507.01760*, 2015.

[19] M. Moakher, "On the averaging of symmetric positive-definite tensors," *Journal of Elasticity*, vol. 82, no. 3, pp. 273–296, 2006.

[20] B. Afsari, "Riemannian lp center of mass: existence, uniqueness and convexity," *Proceedings of the American Mathematical Society*, vol. 139, no. 2, pp. 655–673, 2011.

[21] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Computer Vision ECCV 2006*, Ale Leonardis, Horst Bischof, and Axel Pinz, Eds., vol. 3952 of *Lecture Notes in Computer Science*, pp. 589–600. Springer Berlin Heidelberg, 2006.

[22] "Vision Texture Database," MIT Vision and Modeling Group. Available: http://vismod.media.mit.edu/pub/VisTex.

[23] "Outex Texture Database," Center for Machine Vision Research of the University of Oulu. Available: http://www.outex.oulu.fi/index.php?page=classification.

[24] F. Perronnin, J. Snchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Computer Vision ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios, Eds., vol. 6314 of *Lecture Notes in Computer Science*, pp. 143–156. Springer Berlin Heidelberg, 2010.

[25] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, 2010, pp. 3384–3391.

[26] H. Hajri, I. Ilea, S. Said, L. Bombrun, and Y. Berthoumieu, "Riemannian Laplace distribution on the space of symmetric positive definite matrices," *Entropy*, vol. 18, no. 3, pp. 98, 2016.

[27] I. Ilea, L. Bombrun, C. Germain, and Y. Berthoumieu, "Texture image classification with Riemannian Fisher vectors issued from a Laplacian model," in *submitted to IEEE Image Video and Multidimensional Signal Processing (IVMSP) workshop*, 2016.

Statistical Hypothesis Test for Maritime Pine Forest SAR Images Classification Based on the Geodesic Distance

I. Ilea, L. Bombrun, G. Germain, I. Champion, R. Terebes and M. Borda. *Statistical hypothesis test for maritime pine forest SAR images classification based on the geodesic distance*. In IEEE International Geoscience and Remote Sensing Symposium, pages 3215–3218, 2015

STATISTICAL HYPOTHESIS TEST FOR MARITIME PINE FOREST SAR IMAGES CLASSIFICATION BASED ON THE GEODESIC DISTANCE

Ioana Ilea[1,2], *Lionel Bombrun*[1], *Christian Germain*[1], *Isabelle Champion*[3], *Romulus Terebes*[2], *Monica Borda*[2]

[1]: Université de Bordeaux, Laboratoire IMS, Groupe Signal et Image. {*ioana.ilea, lionel.bombrun, christian.germain* }*@ims-bordeaux.fr*

[2]: Technical University of Cluj-Napoca. {*romulus.terebes, monica.borda* }*@com.utcluj.ro*

[3]: INRA Bordeaux, UMR 1391 ISPA. *champion@bordeaux.inra.fr*

ABSTRACT

This paper introduces a new statistical hypothesis test for image classification based on the geodesic distance. We present how it can be used for the classification of texture image. The proposed method is then employed for the classification of Polarimetric Synthetic Aperture Radar images of maritime pine forests on both simulated data with the PolSARproSim software and real data acquired during the ONERA RAMSES campaign in 2004.

Index Terms— Hypothesis test, SAR, geodesic distance, classification.

1. INTRODUCTION

Texture-oriented analyzes on optical images have proven their efficiency for the classification of maritime pine forest. Various approaches have been considered in the literature such as gray-level co-occurrence matrices (GLCM) [1, 2, 3] and more recently wavelet based approaches [4, 5, 6]. In this paper, we investigate how these methods can be extended to SAR and Polarimetric SAR (PolSAR) data [7].

Multiscale approaches have been found to be effective for many image processing applications. In a classification context, the image is decomposed into a set of wavelet subbands, each of them being modeled by a parametric model. During the last decades, many univariate and multivariate parametric models have been proposed including elliptical models [8, 9] and copula based approaches [10, 11]. Next, for each subband, the estimated parameter vector composes the signature of the image. Once the feature vectors are computed for each texture image, a distance (or at least a divergence) is calculated in order to measure the degree of similarity between two images. The similarity measure which computes the proximity between two images should be intrinsic to the parametric model. A well-known choice is the Kullback-Leibler (KL) divergence. Recently, some authors have proposed to consider the geodesic distance which is the shortest path in the parametric manifold. This latter has shown superior retrieval rate compared to the KL divergence for texture image classification [9]. Inspired from previous works on the KL divergence and on the family of (h, ϕ) divergences [12, 13], we introduce a new statistical hypothesis test based on the geodesic distance which is the main objective of the paper. A second contribution concerns the use of a SAR scenes simulator [14] to study the influence of the acquisition parameters (incidence angle, spatial resolution) on classification accuracy.

The paper is structured as follows. Section 2 introduces the proposed statistical hypothesis test based on the geodesic distance. Section 3 presents an application for the classification of pine forests based on Polarimetric Synthetic Aperture Radar (PolSAR) images. Classification results are then discussed in Section 4 on both synthetic and real datasets. Conclusions and future works are finally reported in Section 5.

2. STATISTICAL HYPOTHESIS TEST FOR SAR IMAGE CLASSIFICATION

In this paper, we propose to set up a statistical hypothesis test. Let $\chi_1 = (\mathrm{x}_1^1, \ldots, \mathrm{x}_m^1)$ and $\chi_2 = (\mathrm{x}_1^2, \ldots, \mathrm{x}_n^2)$ be two sets of m and n independent and identically distributed random variables (vectors) x according to the parametric models $p(\mathrm{x}|\theta_1)$ and $p(\mathrm{x}|\theta_2)$. Let $\hat{\theta}_1$ and $\hat{\theta}_2$ be the maximum likelihood estimators computed on these sets. In a classification problem, the aim is to determine if χ_1 and χ_2 are issued from the same parametric model. Let's consider the following hypothesis test [13]

$$\begin{cases} H_0 : \theta_1 = \theta_2, \\ H_1 : \theta_1 \neq \theta_2. \end{cases} \tag{1}$$

When $\theta_1 = \theta_2$, we can prove that the statistic $S_{GD}(\widehat{\theta_1}, \widehat{\theta_2}) = \dfrac{mn}{m+n} GD^2(\widehat{\theta}_1, \widehat{\theta}_2)$ is asymptotically chi-square distributed with M degrees of freedom for sufficiently large value of m and n. The degree of freedom M is equal to the dimension of the parameter space ($M = d(d+1)/2$). In the following, we propose an application to the zero-mean multivariate Gaussian distribution (MGD). In such case, the geodesic distance is given by $GD(\hat{\mathrm{M}}_1, \hat{\mathrm{M}}_2) = \left[\frac{1}{2}\sum_i (\ln \lambda_i)^2\right]^{\frac{1}{2}}$, where $\hat{\mathrm{M}}_1$ and $\hat{\mathrm{M}}_2$ are the maximum likelihood estimates of two MGDs covariance matrices and λ_i, $i = 1 \ldots d$ are the eigenvalues of $\hat{\mathrm{M}}_2^{-1}\hat{\mathrm{M}}_1$.

3. APPLICATION TO MARITIME PINE FOREST CLASSIFICATION

The dataset used for this work contains both simulated and real L-band SAR images. First, the simulated dataset is used to determine the best airborne configuration for maritime pine classification according to the stand age. In other words, is it better to have a single high resolution SAR image or a low resolution PolSAR image with two or three channels? Second, real SAR images are used to validate the results.

3.1. Database

3.1.1. Simulated L-band SAR images

The simulated dataset is created by using the PolSARproSim software. This software provides fully polarized simulated SAR images of forest displaying properties consistent with real SAR imagery [14]. Images are obtained by specifying various acquisition parameters such as the platform altitude, the incidence angle, the frequency, the azimuth and slant range resolutions, and some forest stand properties, including the stand area and density, the tree species and their mean height.

For our study, pine tree forests of 5, 6, 12, 15, 21, 25 and 32 years old are simulated. The platform altitude is set to 3580 meters, corresponding to an airborne system, while the frequency is fixed at 1.3 GHz (L-band). In order to find the best airborne configuration, two experiments are considered. In the first case, the incidence angle is chosen to be 45° and the influence of the spatial resolution on classification performance is evaluated. Five datasets are simulated at a resolution of respectively 0.5, 1, 2, 3 and 5 meters. In the second case, the image resolution is fixed to 0.5 meters and several incidence angles are tested: 25°, 35°, 45° and 55°.

In both cases, the stand density (D) and the mean tree height ($\bar{H}$) are set according to the desired stand age, as mentioned in Table 1.

Age	5	6	12	15	21	25	32
$\bar{H}$	5.5	6.5	11.6	13.7	17.3	19.2	21.9
D	1200	1200	800	800	400	400	300

Table 1: Maritime pine stand density D (stems/ha) and mean tree height $\bar{H}$ (meters) as a function of stand age (years).

The values for the stand density are chosen to be equal to those given by the *Centre Régional de la Propriété Forestière Aquitaine*, France for the maritime pine, while the mean tree height $\bar{H}$ is obtained by using the Maugé theoretical model [15] given by $\bar{H} = H_{max}(1 - 0.96^a)$, where $H_{max} = 30$ meters is the maximum height and a is the stand age.

By using these numerical values, a database of 350 images is created for each experiment and structured in 4 classes, according to the stand age: 1^{st} **class:** less than 10 years (Fig. 1(a)); 2^{nd} **class:** between 10 and 20 years (Fig. 1(b)); 3^{rd} **class:** between 20 and 30 years (Fig. 1(c)); 4^{th} **class:** over 30 years (Fig. 1(d)).

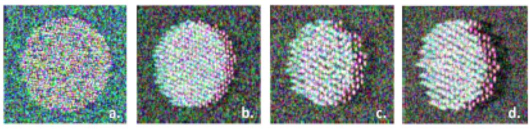


Fig. 1: Examples of L-band pine forest images of: (a) 5, (b) 15, (c) 21 and (d) 32 years old simulated with PolSARproSim software for an incidence angle of 45° and a resolution of 1 meter.

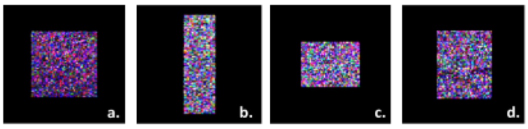


Fig. 2: The real L-band SAR image and examples of pine forest stands of: 5 (a), 15 (b), 21 (c) and 32 (d) years old.

3.1.2. Real L-band SAR image

The real L-band SAR data displayed in Fig. 2 consists in one fully polarimetric image (1 meter resolution) acquired on the Nezer maritime pine forest in France, during an ONERA RAMSES campaign in 2004. From this image, 62 forest stands between 5 and 48 years old are identified and grouped in 4 classes, as it was done for the simulated images.

In the next section, we present several strategies for modeling SAR images and hence obtaining the corresponding feature vectors.

3.2. Methodology

3.2.1. GLCM

The GLCMs are computed on the real-valued HV polarization image transformed in dB and quantified with 32 gray levels. The number of quantization levels is chosen by taking into consideration the image size. In a Cartesian coordinate system, the GLCMs are functions of two parameters: the distance d between neighboring pixels and the direction α. For our study, d varies between 1 and 15, and $\alpha = \{0°, 45°, 90°, 135°\}$. The Haralick [1] textural descriptors *homogeneity*, *entropy*, and *correlation* along with the *GLCM mean* are extracted and averaged in the four directions to reduce the sensibility to the stand's orientation [6]. Further on, this method is denoted by *GLCM HV*.

3.2.2. MGD model for a single polarization image

The real-valued HH polarization image transformed in dB is decomposed by using a Daubechies 4 (db4) wavelet transform, with 2 levels and 3 orientations. For each subband, a spatial dependence with a 3 × 3 neighborhood is considered and modeled by the MGD. The parameter of this distribution is estimated by the Sample Covariance Matrix (SCM). In the following, this algorithm is denoted by *MGD HH + WT + S*.

3.2.3. MGD model for a three polarization image

The HH, HV and VV polarization images are merged into a 3-dimensional array, with each pixel being a complex number. Three different algorithms are developed based on:

- *the polarimetric dependence* (denoted *MGD Polar*): the complex 3-dimensional array is modeled by the MGD and a 3 × 3 covariance matrix is estimated with the SCM algorithm.
- *the polarimetric dependence and the wavelet decomposition* (denoted *MGD Polar + TW*): the complex 3-dimensional array is filtered using the db4 wavelet transform with 2 levels and 3 orientations. Each subband is modeled by the MGD and the 3 × 3 covariance matrix is estimated with the SCM algorithm.
- *the polarimetric and spatial dependence, along with the wavelet decomposition* (denoted *MGD Polar + TW + S*): the complex 3-dimensional array is decomposed using a db4 wavelet transform having 2 levels and 3 orientations. For each subband, a spatial dependence given by a 3 × 3 neighborhood is modeled by the MGD and a 27 × 27 covariance matrix is estimated with the SCM algorithm.

4. RESULTS

In the context of a supervised classification, the database is randomly divided into a training and a testing set by a cross-validation procedure. The partitioning algorithm is repeated 100 times and for each iteration half of the database is used for training, while the other half is used for testing. Once the feature vector extracted for all images, a similarity measure between testing and training images is computed by using the Mahalanobis distance for the GLCM algorithm and the statistic S_{GD} defined in Section 2 for the others. All the previously described algorithms are tested and the retrieval performance is evaluated by means of the overall accuracy computed for a k Nearest Neighbor classifier (k-NN), with k set to 5. In the following, the classification performances obtained on both simulated and real SAR images are presented.

4.1. Simulated L-band SAR images

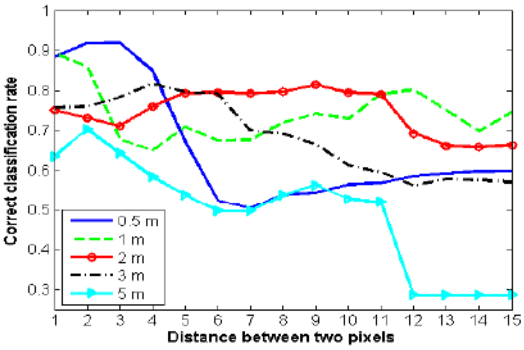


Fig. 3: Influence of distance d in GLCM on classification accuracy for different spatial resolution (HV channel).

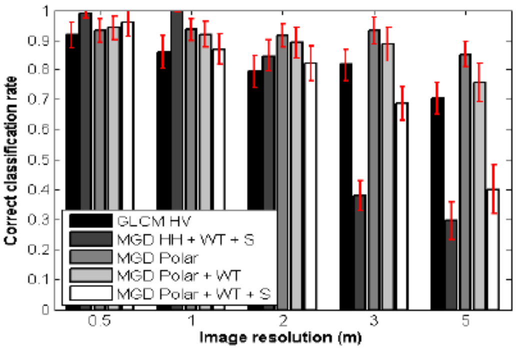


Fig. 4: Influence of the spatial resolution on classification accuracy for simulated L-band SAR images with incidence angles of 45°.

As mentioned in Section 3.1.1, two types of experiments are performed on simulated data. First, the influence of the image resolution is tested. For this experiment the incidence angle is fixed to 45° and the image resolution varies from 0.5m to 5m. Fig. 3 draws the influence of distance d to find the best distance between neighboring pixels. It can be seen that distances between 1 and 5 pixels give the best results. Fig. 4 shows a comparison between the GLCM algorithm and the statistical based approaches with the geodesic distance, knowing that each time the polarization with the best performance is retained. As observed in Fig. 4, for simulated data it is better to have one very high resolution polarization channel (99 ± 1% for MGD HH + WT + S at 0.5 meters) than a low resolution fully polarimetric SAR image (85 ± 4.5% for MGD Polar at 5 meters). For this example, a significant gain of about 14 points is observed.

Second, the influence of the incidence angle is analyzed. For this experiment, the image resolution is fixed to 0.5m and several incidence angles are considered. Like in the previous case, tests are performed to find the appropriate distance d for the GLCM algorithm. The best classification rates are retained and compared in Fig. 5 with those given by the statistical based methods. As it can be seen, the GLCM HV is influenced by the incidence angle, while some small changes can be spotted for the other methods.

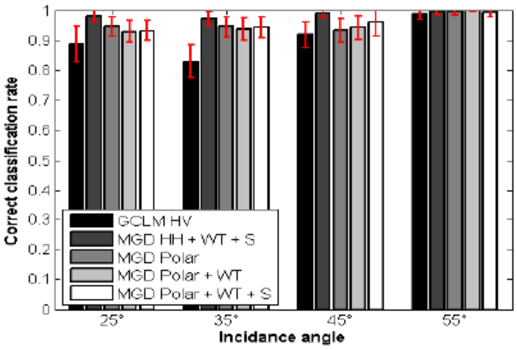


Fig. 5: Influence of the incidence angle on classification accuracy for simulated L-band SAR images having a resolution of 0.5 meter.

4.2. Real L-band SAR images

Even though PolSARproSim provides a high level of realism, significant differences can be observed between simulated (Fig. 1) and real data (Fig. 2). Those differences are the results of various phenomena, such as forest management practices (thinning operations, plantation density) and natural hazards (storm damages), yielding to some within-class diversity. Hence, as displayed in Table 2, classification results on real SAR images are lower to those shown in Section 4.1 on synthetic dataset. Similar to the case of simulated images with a resolution of 3 and 5m, the best results are given for GLCM HV (86.6 ± 5.6%) and MGD Polar (84.0 ± 4.4%) methods.

Classification method	Overall accuracy
GLCM HV	**86.6 ± 5.6**
MGD HH + WT + S	59.0 ± 5.4
MGD Polar	**84.0 ± 4.4**
MGD Polar + WT	81.8 ± 4.0
MGD Polar + WT + S	63.5 ± 4.9

Table 2: Comparison between the classification algorithms for real L-band SAR images.

5. CONCLUSION

In this paper, we have introduced a statistical hypothesis test based on the geodesic distance. Various experiments have been conducted on both simulated and real SAR data for the classification of maritime pine forest images. Experiments on simulated dataset have shown that it is better to have one very high resolution polarization channel than a low resolution fully polarimetric SAR image. Due to the presence of intra-class diversity, those conclusions are slightly modified on real SAR data.

Further works will include the generalization of the proposed hypothesis test to robust estimators such as the family of M-estimators [16].

6. REFERENCES

[1] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. on Syst., Man and Cyber.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[2] F. Kayitakire, C. Hamel, and P. Defourny, "Retrieving forest structure variables based on image texture analysis and IKONOS-2 imagery," *Rem. Sens. of Env.*, vol. 102, no. 3, pp. 390–401, 2006.

[3] I. Champion, P. Dubois-Fernandez, D. Guyon, and M. Cottrel, "Radar image texture as a function of forest stand age," *International Journal of Remote Sensing*, vol. 29, no. 6, pp. 1795–1800, 2008.

[4] M.N. Do and M. Vetterli, "Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance," *IEEE Trans. on Im. Process.*, vol. 11, pp. 146–158, 2002.

[5] O. Regniers, L. Bombrun, D. Guyon, J.-C. Samalens, C. Tinel, G. Grenier, and C. Germain, "Wavelet based texture modeling for the classification of very high resolution maritime pine forest images," in *IEEE IGARSS*, July 2014, pp. 2027–2030.

[6] O. Regniers, L. Bombrun, D. Guyon, J.-C. Samalens, and C. Germain, "Wavelet-based texture features for the classification of age classes in a maritime pine forest," *IEEE Geosc. and Rem. Sens. Lett.*, vol. 12, no. 3, pp. 621–625, March 2015.

[7] I. Champion, C. Germain, J. P. Da Costa, A. Alborini, and P. Dubois-Fernandez, "Retrieval of forest stand age from sar image texture for varying distance and orientation values of the gray level co-occurrence matrix," *IEEE Geosc. and Rem. Sens. Lett.*, vol. 11, no. 1, pp. 5–9, 2014.

[8] L. Bombrun, Y. Berthoumieu, N.-E. Lasmar, and G. Verdoolaege, "Mutlivariate texture retrieval using the geodesic distance between elliptically distributed random variables," in *IEEE ICIP*, 2011, pp. 3637–3640.

[9] G. Verdoolaege and P. Scheunders, "On the geometry of multivariate generalized Gaussian models," *Journal of Math. Imag.and Vis.*, vol. 43, no. 3, pp. 180–193, 2012.

[10] R. Kwitt and A. Uhl, "Lightweight probabilistic texture retrieval," *IEEE Trans. on Im. Process.*, vol. 19, no. 1, pp. 241–253, 2010.

[11] Y. Stitou, N.-E. Lasmar, and Y. Berthoumieu, "Copulas based multivariate gamma modeling for texture classification," in *IEEE ICASSP*, 2009, pp. 1045–1048.

[12] M. Salicru, D. Morales, M.L. Menendez, and L. Pardo, "On the applications of divergence type measures in testing statistical hypotheses," *Journal of Multivariate Analysis*, vol. 51, no. 2, pp. 372 – 391, 1994.

[13] A.D.C. Nascimento, R.J. Cintra, and A.C. Frery, "Hypothesis testing in speckled data with stochastic distances," *IEEE Trans. on Geosc. and Rem. Sens.*, vol. 48, pp. 373–385, 2010.

[14] M.L. Williams, "PolSARproSim: A Coherent, Polarimetric SAR Simulation of Forests for PolSARPro. Design Document ans Algorithm Specification (u1.0) ," 2006.

[15] J.P. Maugé, *Le pin maritime premier résineux de France*, Centre de Productivité et d'Action Forestière d'Aquitaine, Institut pour le Développement Forestier, Paris, 1987.

[16] I. Ilea, L. Bombrun, C. Germain, R. Terebes, and M. Borda, "Statistical hypothesis test for robust classification on the space of covariance matrices," in *IEEE ICIP*, 2015.