



**HAL**  
open science

# Contributions à la segmentation non supervisée d'images hyperspectrales : trois approches algébriques et géométriques

Saadallah El Asmar

## ► To cite this version:

Saadallah El Asmar. Contributions à la segmentation non supervisée d'images hyperspectrales : trois approches algébriques et géométriques. Traitement du signal et de l'image [eess.SP]. Université de La Rochelle, 2016. Français. NNT : 2016LAROS023 . tel-01661468

**HAL Id: tel-01661468**

**<https://theses.hal.science/tel-01661468>**

Submitted on 12 Dec 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.



UNIVERSITÉ DE LA ROCHELLE

## THÈSE

Pour l'obtention du grade de  
**DOCTEUR** de l'**UNIVERSITÉ DE LA ROCHELLE**

École Doctorale : **Sciences et Ingénierie pour l'Information, Mathématiques (S2IM)**

Laboratoire : **Mathématiques, Image et Applications (MIA)**

Mention : **MATHÉMATIQUES ET APPLICATIONS**

Présentée par

**Saadallah EL ASMAR**

---

**Contributions à la segmentation non supervisée d'images  
hyperspectrales : trois approches algébriques et géométriques**

---

soutenue le 30 Août 2016  
devant la commission d'Examen :

### Jury

M. Yannick BERTHOUMIEU	Rapporteur
M. Mohamad KHALIL	Rapporteur
M. Philippe CARRÉ	Examineur
M. Alamin MANSOURI	Examineur
M. Michel BERTHIER	Directeur de thèse
M. Carl FRÉLICOT	Directeur de thèse



# Table des matières

<b>1</b>	<b><u>Introduction</u></b>	<b>9</b>
1.1	L'acquisition d'images hyperspectrales . . . . .	10
1.2	L'analyse d'images hyperspectrales . . . . .	12
1.3	Trois propositions pour la segmentation . . . . .	13
1.3.1	Segmentation par similarité . . . . .	14
1.3.2	Segmentation par codage binaire . . . . .	14
1.3.3	Segmentation riemannienne . . . . .	16
1.4	Implémentation et évaluation . . . . .	17
1.4.1	Algorithmes $k$ -means . . . . .	18
1.4.2	Rand Index . . . . .	20
<b>2</b>	<b><u>Segmentation par similarité</u></b>	<b>23</b>
2.1	Similarité spectrale entre pixels . . . . .	24
2.1.1	Appariements . . . . .	24
2.1.2	Prise en compte du bruit . . . . .	25
2.2	Segmentations locales . . . . .	27
2.2.1	Un exemple synthétique . . . . .	27
2.2.2	L'image "Pavia University Scene" . . . . .	29
2.2.3	Sélection de bandes . . . . .	30
2.3	Similarité spectrale entre objets . . . . .	32
2.4	Segmentation globale . . . . .	34
2.4.1	Des segmentations locales à la segmentation globale . . . . .	34
2.4.2	L'image "Pavia University Scene" . . . . .	36
2.4.3	D'autres exemples . . . . .	37
2.5	Appendice . . . . .	38
2.6	Conclusions et perspectives du chapitre . . . . .	40
<b>3</b>	<b><u>Segmentation par codage binaire</u></b>	<b>43</b>
3.1	Codage binaire de variations spectrales . . . . .	44
3.1.1	Visualisation . . . . .	45
3.1.2	Exemples de visualisation directe . . . . .	45
3.1.3	Segmentation riemannienne . . . . .	45
3.1.4	Exemples de détection de contours . . . . .	47
3.2	Distance de Hamming généralisée . . . . .	48
3.2.1	Définition de la distance . . . . .	49



3.2.2	Calcul de la distance . . . . .	50
3.2.3	Un exemple simple . . . . .	52
3.3	Segmentation par clustering . . . . .	52
3.3.1	Algorithmes $k$ -modes . . . . .	53
3.3.2	Illustrations . . . . .	55
3.3.3	Amélioration du codage binaire . . . . .	55
3.3.4	Comparaison avec des méthodes de référence . . . . .	58
3.4	Conclusions et perspectives du chapitre . . . . .	59
<b>4</b>	<b><u>Segmentation riemannienne</u></b> . . . . .	<b>63</b>
4.1	Images hyperspectrales et variétés . . . . .	64
4.2	La géométrie de l'espace $SPD(d)$ . . . . .	66
4.2.1	$SPD(1)$ et le demi-plan de Poincaré . . . . .	66
4.2.2	$SPD(d)$ et le demi-espace de Siegel . . . . .	67
4.3	Moyennes et médianes dans $SPD(d)$ . . . . .	69
4.3.1	Barycentre de Karcher . . . . .	69
4.3.2	Barycentre itératif . . . . .	72
4.3.3	Médiane . . . . .	73
4.4	Classification et segmentation . . . . .	75
4.4.1	Clustering dans $SPD(2)$ . . . . .	76
4.4.2	Clustering dans $SPD(2)^{\Lambda}$ . . . . .	78
4.4.3	Clustering dans $SPD(3)$ . . . . .	80
4.5	Conclusions et perspectives du chapitre . . . . .	81
<b>5</b>	<b><u>Conclusion</u></b> . . . . .	<b>85</b>
5.1	Résumé des travaux de la thèse . . . . .	85
5.2	Comparaisons des trois approches . . . . .	85
5.3	Perspectives . . . . .	86

# Table des figures

1	Deux exemples de profils spectraux issus de Aster Spectral Library . . . . .	10
2	Le principe d'acquisition d'images hyperspectrales par télédétection . . . . .	11
3	Exemples de courbes de réflectance spectrale . . . . .	11
4	Image PaviaU ( $\lambda = 30$ ) et deux imagettes patches . . . . .	15
5	Exemples de visualisation directe à l'aide de codages binaires. . . . .	16
6	Comparaison entre deux algorithmes de calcul de moyennes de Karcher (par descente de gradient, DG, et de façon récrusive, REC). . . . .	18
7	Les $\Lambda = 14$ bandes (de bas en haut et de gauche à droite) d'une imagette synthétique bruitée avec $\sigma = 0,1$ . . . . .	27
8	Évaluation des résultats de segmentation locale pour l'imagette synthétique : la courbe du Rand Index en fonction de $\varepsilon$ pour différentes valeurs de $\sigma$ . . . . .	28
9	Segmentation locale de l'imagette synthétique pour différentes valeurs de $\varepsilon$ , avec $\eta = 4$ et $\sigma = 0,1$ . . . . .	28
10	Segmentation locale de l'imagette synthétique pour différentes valeurs du paramètre de robustesse $\eta$ , avec $\varepsilon = 0,08$ et $\sigma = 0,1$ . . . . .	29
11	PaviaU (de gauche à droite) : la vérité-terrain de l'image $I$ ( $610 \times 340 \times 103$ ), l'image $I(30)$ , deux imagettes $I_1(30)$ (en haut) et $I_2(30)$ (en bas), et leur vérité-terrain. . . . .	30
12	Évaluation de la segmentation locale pour les deux imagettes test de PaviaU : à gauche les classes de la vérité terrain, à droite le résultat de la segmentation. . . . .	31
13	Segmentation de l'imagette test $I_1$ de PaviaU avec $\eta = 30$ et $\varepsilon = 0,0051$ (Rand Index=0,9833). . . . .	31
14	Segmentations locales et globale de l'image PaviaU . . . . .	36
15	Segmentation locale de l'image PaviaU . . . . .	37
16	Segmentation globale de l'image PaviaU . . . . .	38
17	Segmentation de l'image Salinas . . . . .	39
18	Segmentation de l'image Indiana . . . . .	40
19	Approche pyramidale de la segmentation . . . . .	41
20	Conversion décimale du codage binaire pour les images Flowers et Balloons : RGB (à gauche), $I_b^+$ (au centre), $I_b^-$ (à droite). . . . .	46
21	Conversion décimale du codage binaire pour l'image PaviaU. . . . .	47
22	Conversion décimale et détection de contours : image $I_b^+$ (à gauche), indicateur de force $\varpi$ (au centre) et contours détectés (à droite). . . . .	48

23	Exemples de résultats de l'algorithme de $k$ -modes avec $k = 5$ et deux valeurs différentes pour le coût du shift $\alpha_S$ : Flowers (première ligne) et Balloons (seconde ligne).	56
24	Exemples de segmentation par $k$ -modes sur l'image PaviaU pour différentes valeurs de $k$ et le paramètre du shift $\alpha_S$ fixé à 0,2.	57
25	Exemples de segmentation par $k$ -modes sur l'image PaviaU pour différentes valeurs du paramètre du shift $\alpha_S$ et $k$ fixé à 6.	57
26	Resultats obtenus à partir de l'algorithme de $k$ -modes modifié avec $k = 10$ et deux valeurs différentes du paramètre $\delta$ : (a) and (b). La bande échantillon de "PaviaU" et la tache détectée : (c) and (d).	59
27	Segmentation de l'image PaviaU par des méthodes de la littérature.	60
28	Segmentation de l'image PaviaU avec $k = 10$ et $\alpha_S = 0.2$ et application d'un filtre médian de différentes tailles.	60
29	Représentation schématique de la segmentation par clustering dans $SPD(2)$ .	75
30	Comparaison entre les deux algorithmes de calcul de moyennes de Karcher.	76
31	Clustering dans $SPD(2)$ des images Balloons et Flowers.	77
32	Clustering dans $SPD(2)$ de l'image PaviaU.	77
33	Clustering avec médiane dans $SPD(2)$ des images Balloons et Flowers.	78
34	Clustering avec médiane dans $SPD(2)$ de l'image PaviaU.	79
35	Les effets des filtres médians dans $SPD(2)$ .	79
36	Comparaisons des résultats de clustering dans $SPD(2)^A$ et $SPD(2)$ sur l'image PaviaU avec $k = 6$ .	80
37	Clustering dans $SPD(3)$ des images Balloons et Flowers.	81
38	Clustering dans $SPD(3)$ de l'image PaviaU.	82
39	Comparaison des trois clusterings sans filtres médians sur l'image PaviaU avec $k = 6$ .	82
40	Clustering dans $SPD(3)$ de l'image PaviaU sans et avec la permutation spectrale – Les paramètres d'initialisation sont identiques et $k = 7$ .	83
41	Exemples de segmentation par l'agorithme de $k$ -modes flou sur l'image PaviaU pour différentes valeurs du paramètre de flou $\beta$ , $\alpha_S = 0.3$ et $k = 7$ .	88
42	Exemples de segmentation par l'agorithme de $k$ -means flou dans $SPD(3)$ sur l'image PaviaU pour différentes valeurs du paramètre de flou $\beta$ et $k = 7$ .	88

# Liste des tableaux

- 1 PaviaU : les classes de la vérité-terrain. . . . . 30
- 2 Maxima des moyennes des réponses spectrales des objets de l'imagette synthétique. . . . . 32
- 3 Les bandes spectrales significatives des objets de l'imagette synthétique. . . . . 32
- 4 Les plages de bandes spectrales significatives de l'image PaviaU . . . . . 37
- 5 Tableau de comparaisons entre les trois approches proposées. . . . . 86



# Chapitre 1

## Introduction

Nous nous intéressons dans cette thèse au problème de la segmentation non supervisée d'images hyperspectrales. Par définition, des données hyperspectrales sont des données qui sont rassemblées dans un nombre  $\Lambda$  de bandes spectrales, avec  $\Lambda > 1$  [42]. Il n'existe pas vraiment de consensus quant à la valeur de  $\Lambda$  qui permet de différencier les images multispectrales des images hyperspectrales. Typiquement, une image couleur est une image multispectrale avec  $\Lambda = 3$ . Depuis environ une dizaine d'années, les images hyperspectrales produites par les systèmes de télédétection, "Remote Sensing", ont permis d'obtenir des informations très fiables quant aux caractéristiques spectrales de matériaux présents dans une scène donnée. Les spectromètres actuels fournissent des images hyperspectrales correspondant à des centaines de bandes presque contiguës couvrant un large domaine du spectre magnétique. Il est communément admis que dans la pratique une analyse hyperspectrale nécessite de travailler avec au moins dix bandes mais parfois aussi avec beaucoup plus. Le traitement de ces images permet notamment de faire de la détection d'anomalies dans une scène hyperspectrale en exploitant les signatures des matériaux, comme par exemple dans [48]. Les améliorations constantes en termes de résolution spectrale nécessitent de développer des algorithmes originaux de traitement et d'exploitation du signal et de l'information. Bien que les images hyperspectrales ne soient pas statistiquement très différentes des images en niveau de gris ou en couleur [11], leur analyse est beaucoup plus compliquée. Ceci est dû en particulier à leur volume très conséquent mais également à d'autres caractéristiques comme la redondance spectrale ou la nature du bruit d'acquisition. La plupart des méthodes d'analyse d'images hyperspectrales sont dites "pixelliques" : elles utilisent les informations spectrales associées à des "cellules" de l'image. Peu d'approches abordent le problème en considérant les variations spatiales à spectres ou groupes de spectres donnés. Nous mentionnons dans la section 1.2 (page 12) quelques-unes des méthodes d'analyse les plus populaires.

L'origine de notre travail est une question assez naive : dans quelle mesure peut-on étendre les algorithmes de traitement et d'analyse d'images en niveau de gris ou couleur aux images hyperspectrales en ayant le moins d'*a priori* possible sur la signification ou les caractéristiques des profils spectraux ? Avec ce point de vue, trois stratégies au moins sont envisageables. La première consiste à considérer qu'une image hyperspectrale n'est rien d'autre qu'une image à  $\Lambda$  canaux avec  $\Lambda$  grand. Dans ce cas, la donnée d'un profil spectral est équivalente à la donnée d'un vecteur à coordonnées positives de  $\mathbb{R}^\Lambda$ . C'est de cette façon que l'on aborde le problème de la segmentation au moyen de mesures de similarité dans le

chapitre 2, page 23. La deuxième stratégie consiste à appréhender une image hyperspectrale comme une image de courbes spectrales. Dans ce cas, la donnée d’un profil spectral est équivalente à la donnée d’une courbe, plus précisément le graphe de la réflectance comme fonction de la longueur d’onde. Les résultats de segmentation du chapitre 3 (page 43) sont obtenus à partir d’un algorithme de classification reposant sur un codage binaire des variations de ces courbes. La troisième stratégie consiste à décrire une image hyperspectrale comme au choix une surface, une famille de surfaces ou une variété riemannienne de dimension trois. À chacune de ces variétés correspond une métrique, c’est-à-dire un champ de tenseurs symétriques définis positifs. On associe donc à chaque pixel de l’image au choix un élément de  $SPD(2)$ ,  $\Lambda$  éléments de l’ensemble  $SPD(2)$  ou un élément de  $SPD(3)$  où  $SPD(d)$  désigne l’ensemble des matrices  $d \times d$  symétriques définies positives. Dans le chapitre 4 (voir page 63), nous proposons un algorithme de classification suivant cette approche. Il est souligné dans [42] : “*Again assuming that the data were gathered in a larger number of bands than is necessary or desirable for the particular analysis at hand, an important early step is to form the feature subset that is to be used in the analysis*”. D’une certaine façon, nous testons trois espaces de caractéristiques simples : l’espace  $\mathbb{R}^\Lambda$ , ou après renormalisation l’espace  $[0, 1]^\Lambda$ , l’espace des graphes au-dessus de l’axe des longueurs d’ondes, ou après codage binaire des variations l’espace  $\{0, 1\}^{\Lambda-1}$  et l’espace  $SPD(d)$ .

## 1.1 L’acquisition d’images hyperspectrales

La figure 1 montre deux courbes spectrales<sup>1</sup> correspondant à deux matériaux manufacturés. Dans ce cas précis les données sont très volumineuses. Il peut être bon de rappeler en

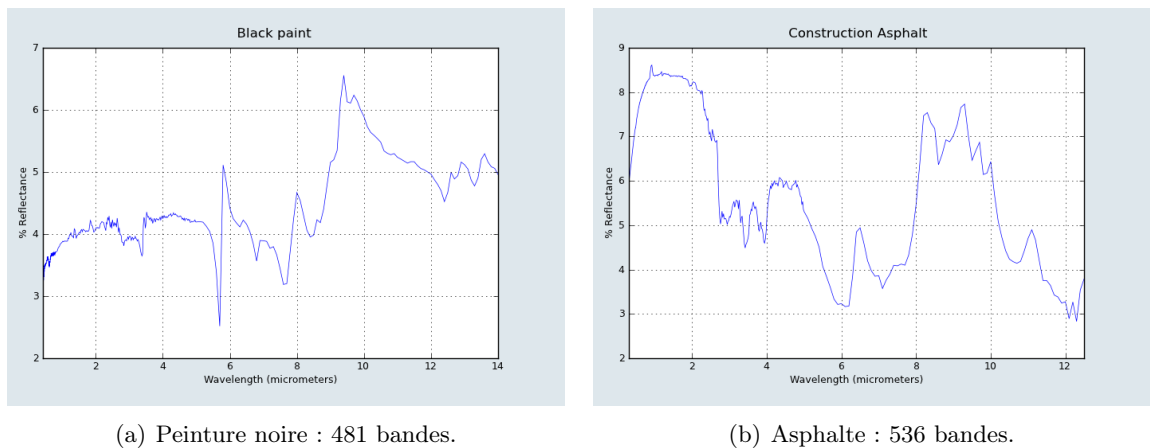


FIGURE 1 – Deux exemples de profils spectraux issus de Aster Spectral Library

préambule la façon avec laquelle on obtient ce type de profils. D’une façon générale, la spectroscopie a pour but de mesurer les variations, en fonction des longueurs d’onde, de l’énergie de la lumière émise ou réfléchié par un matériau. Dans le cas de la télédétection (*Remote Sensing*), la lumière est celle du soleil et les matériaux, manufacturés ou naturels, ceux présents à la surface de la terre. Des spectromètres de laboratoire sont utilisés pour

1. Voir : <http://speclib.jpl.nasa.gov/>

caractériser la lumière réfléchiée par un matériau test. La lumière est décomposée, à l'aide par exemple de prismes, suivant de nombreuses longueurs d'onde très proches et l'énergie correspondant à chacune de ces longueurs d'onde est mesurée par un détecteur dédié à cette longueur d'onde. Dans la pratique, les longueurs d'onde sont généralement séparées par un intervalle de 0,01 micromètre et couvrent un domaine spectral allant de 0,4 à 2,4 micromètres, du visible à l'infrarouge moyen (voir la figure 2 pour une illustration<sup>2</sup>).

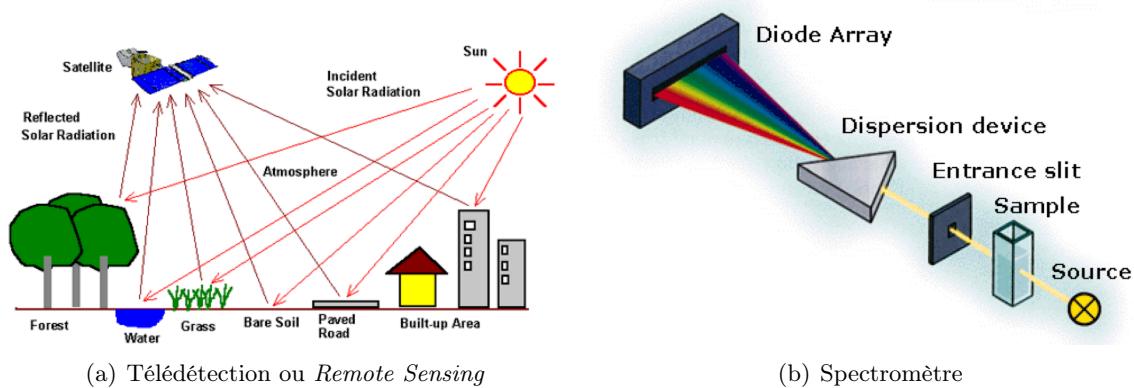


FIGURE 2 – Le principe d'acquisition d'images hyperspectrales par télédétection

L'information principale que l'on acquiert par ce principe est la réflectance spectrale, c'est-à-dire le rapport entre l'énergie réfléchiée et l'énergie incidente comme fonction de la longueur d'onde. La réflectance varie en fonction de la longueur d'onde pour la plupart des matériaux, les quantités d'énergie réfléchiée et d'énergie absorbée pour chacune des longueurs d'onde étant caractéristiques des matériaux. Le postulat de base est que les courbes de réflectance spectrale sont discriminantes comme illustré à la figure 3.

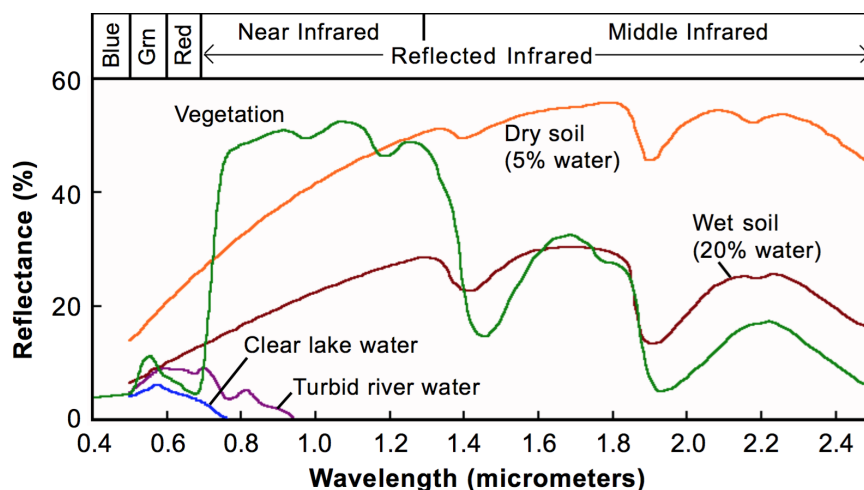


FIGURE 3 – Exemples de courbes de réflectance spectrale

La qualité de l'information de réflectance spectrale dépend de la résolution spatiale. Si les

2. Les images sont tirées de sites des universités de San Diego et de New Mexico



cellules spatiales sont grandes, plusieurs matériaux peuvent contribuer lors de la mesure. On obtient alors un spectre mélangé constitué de spectres purs, les *endmembers*.

Nous renvoyons le lecteur par exemple à l'ouvrage [59] ou au numéro spécial [46] pour plus de détails.

## 1.2 L'analyse d'images hyperspectrales

Il est impossible de faire un tour d'horizon complet des diverses méthodes d'analyse d'images hyperspectrales. Nous nous bornons ici à décrire très brièvement trois parmi les plus répandues des approches. Nous nous inspirons de l'état de l'art récent [46].

1. Les algorithmes de détection [48, 49, 52, 55]. Le but des algorithmes de détection d'anomalies est d'identifier des pixels d'une scène donnée dont la signature spectrale, connue *a priori* ou non suivant le contexte, diffère considérablement des signatures spectrales des pixels voisins. Il s'agit donc essentiellement d'élaborer des classifieurs binaires qui associent à chaque pixel une étiquette de cible ou de fond. Nombre de ces classifieurs, et parmi ceux-ci le RX-algorithm, reposent sur une modélisation statistique du fond à l'aide par exemple de distributions gaussiennes multivariées dont il s'agit d'estimer les moyennes et les covariances, voir par exemple [63]. Des variantes utilisent des décompositions en composantes principales des matrices de covariance des modèles ([65]). Cependant, en général, le fond se prête mal à une telle description par des gaussiennes multivariées (ou même des mélanges de telles gaussiennes). Des approches alternatives sont basées sur des caractérisations à l'aide de distributions *t*-ECD (*t-Elliptical Contoured Distributions* [47]) ou sur des descriptions non linéaires par noyaux (KRX-algorithm, [40]).
2. La classification d'images hyperspectrales. Dans la communauté de la télédétection, le terme classification désigne un procédé qui attribue à chaque pixel d'une image hyperspectrale une classe d'un ensemble de classes. Le terme segmentation désigne quant à lui un procédé qui regroupe des pixels en objets, ces derniers étant ensuite attribués à des classes [12]. Les méthodes les plus récentes font la part belle à l'apprentissage. L'apprentissage statistique [23, 74] est une théorie qui permet d'obtenir, par apprentissage des relations entre données d'entrée et données de sortie, des fonctions linéaires qui prédisent de façon fiable quel attribut associer à un échantillon. Ces fonctions sont le plus souvent solutions de problèmes variationnels : elles minimisent une fonctionnelle

$$\mathcal{L}(f) = \mathcal{E}(f) + \nu \mathcal{R}(f) \tag{1}$$

où  $\mathcal{E}$  est un terme d'attache aux données (contrôle de l'erreur) et  $\mathcal{R}$  est un terme de régularisation. Il est à noter que la complexité des données hyperspectrales nécessite bien souvent de privilégier des méthodes non linéaires. Cette remarque justifie par ailleurs une deuxième approche par apprentissage basée sur les représentations par graphes (*nonlinear manifold learning/graph embedding framework*, voir par exemple dans [17]). À un ensemble  $X$  d'échantillons est associé un graphe non orienté  $G = \{X, W\}$  où  $W$  est une matrice de similarité. À partir de cette matrice, plusieurs stratégies de réduction de dimension peuvent être envisagées [45].

3. La séparation de sources [8, 16, 36]. La séparation de sources (*spectral unmixing*), est un moyen de retrouver les composantes élémentaires qui se trouvent mélangées dans les données acquises. Ce procédé permet d'identifier les signatures spectrales des matériaux présents dans une scène donnée, les *endmembers*, ainsi que de déterminer leur contribution individuelle, c'est-à-dire leur abondance. C'est en général un problème mal posé qui donne donc lieu à de multiples solutions admissibles. La séparation de sources est dite supervisée si les caractéristiques spectrales des *endmembers* sont connues, et est dite non supervisée dans le cas contraire. La plupart des approches nécessitent de définir un modèle de mélange qui décrit de façon analytique la manière dont les *endmembers* se combinent pour former le spectre mélangé acquis. Un des modèles les plus simples, qui ne tient compte que partiellement de la complexité des données, est le modèle linéaire LMM (*Linear Mixing Model*). Si  $I$  est une image hyperspectrale à  $\Lambda$  bandes, ce modèle est donné par

$$I(m, n) = \sum_{k=1}^E \alpha_k S_k(m, n) + \beta(m, n) = AS(m, n) + \beta(m, n) \quad (2)$$

où  $I(m, n)$  est le vecteur de composantes  $I(m, n, \lambda)$  pour  $1 \leq \lambda \leq \Lambda$  correspondant au pixel  $(m, n)$ ,  $S_k(m, n)$  est le vecteur de composantes  $S_k(m, n, s)$ ,  $1 \leq s \leq E$ , d'abondance du matériel  $k$  au pixel  $(m, n)$ ,  $A$  est la matrice  $\Lambda \times E$  des *endmembers* et  $\beta(m, n)$  est un bruit additif. La séparation de sources se fait traditionnellement en considérant le problème de minimisation

$$\hat{S}(m, n) = \operatorname{argmin}_{S(m, n) \in \mathcal{S}} \|I(m, n) - AS(m, n)\|_2^2 \quad (3)$$

où  $\mathcal{S}$  désigne l'ensemble des vecteurs à coordonnées positives de  $\mathbb{R}^E$  dont la somme des composantes vaut 1. L'approximation linéaire du mélange fournie par le modèle LMM est pertinente sous deux hypothèses [8] : le mélange se produit à l'échelle macroscopique et les photons qui sont détectés n'interagissent qu'avec un seul matériau. Si ces hypothèses ne sont pas satisfaites, il est alors préférable d'avoir recours à des modèles non linéaires.

Mentionnons par ailleurs que peu de travaux envisagent le problème de l'analyse d'images hyperspectrales sous l'angle de la géométrie. Dans [68], les auteurs proposent une méthode originale pour la visualisation qui consiste, par la résolution d'une équation de Poisson, à produire une image en niveaux de gris dont le gradient est un vecteur de contraste<sup>3</sup> de l'image hyperspectrale originale.

### 1.3 Trois propositions pour la segmentation d'images hyperspectrales

Dans ce travail, nous abordons le problème de la segmentation des images hyperspectrales suivant trois approches distinctes.

---

3. Ce vecteur de contraste reflète les variations de la métrique riemannienne de l'image initiale.

### 1.3.1 Segmentation par similarité (voir chapitre 2, page 23)

La première approche s'apparente à la méthode de classification *Graph Embedding*. L'espace des caractéristiques est l'espace  $[0, 1]^\Lambda$  des vecteurs à  $\Lambda$  composantes réelles toutes comprises entre 0 et 1. A chaque pixel  $(m, n)$  d'une image hyperspectrale  $I$  à  $\Lambda$  bandes spectrales correspond un tel vecteur noté  $I(m, n)$ . Une idée très naturelle pour obtenir une segmentation est de comparer entre eux les vecteurs associés à tous les pixels de l'image. Ceci présuppose de disposer d'une *mesure de similarité* entre ces vecteurs, voir par exemple [3]. La mesure de similarité entre deux pixels ayant des profils spectraux semblables est proche de 1 et la mesure de similarité entre deux pixels ayant des profils spectraux différents est proche de 0. À l'aide d'une mesure de similarité, il est possible d'apparier les pixels d'une image. Un *appariement* est une relation binaire réflexive et symétrique, encore appelée relation de tolérance. Deux pixels sont en relation si et seulement si la mesure de similarité entre ces deux pixels est élevée. On obtient par ce biais un graphe non orienté dont les sommets sont les pixels de l'image et dont les arêtes relient les pixels dont les profils spectraux sont similaires. Les classes d'équivalence de la fermeture transitive de ce graphe fournissent les objets de la segmentation.

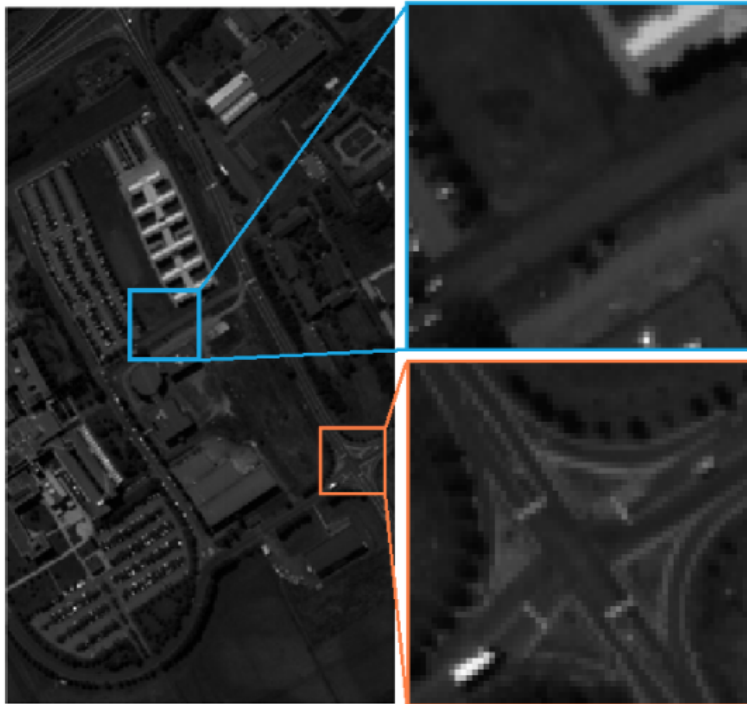
Si cette approche est naturelle, la mise en application n'est pas simple. La première difficulté réside dans la prise en compte du bruit. Pour rendre la relation de tolérance robuste au bruit, il est nécessaire d'introduire une pénalisation. La deuxième difficulté est la complexité algorithmique de cette approche. Cette complexité nous oblige à procéder en deux étapes.

1. Dans un premier temps, nous considérons des *données locales*. Ces données locales sont des images, ou imagerie, hyperspectrales de petite taille qui sont considérées comme des patches d'images hyperspectrales de grande taille (voir la figure 4). Pour ce type de données, la segmentation résulte d'un *appariement des pixels par similarité spectrale*.
2. Dans un second temps, nous procédons à un *appariement des objets* détectés dans chaque imagerie grâce encore une fois à une mesure de similarité spectrale, mais cette fois-ci entre objets.

Le nombre des objets détectés dans chaque imagerie est petit, ce qui permet de réduire de façon importante le coût algorithmique de l'appariement de la seconde étape. Les deux mesures de similarité, entre pixels et entre objets, sont deux paramètres de la méthode. Nous nous sommes volontairement limités à un seul exemple de deux telles mesures et n'avons pas testé les multiples choix que l'on peut envisager, et plus particulièrement les extensions floues. Nous avons en effet préféré tester deux autres approches très sensiblement différentes.

### 1.3.2 Segmentation par codage binaire (voir chapitre 3, page 43)

La seconde approche s'apparente aux méthodes dites de *Spectral Hashing* ou de *Semantic Hashing* utilisées pour la recherche par le contenu ou *information retrieval* (voir par exemple [75]). L'idée générale est d'obtenir un codage binaire de l'information contenue dans une image hyperspectrale et d'analyser cette information à l'aide d'une distance significative en termes de segmentation. L'espace des caractéristiques est dans ce cas l'espace  $\{0, 1\}^{\Lambda-1}$  où  $\Lambda$  est le nombre de bandes spectrales de l'image considérée. Essentiellement trois outils sont mis en œuvre dans cette approche.

FIGURE 4 – Image PaviaU ( $\lambda = 30$ ) et deux imageries patches

1. Le premier est le *codage binaire des variations des courbes spectrales*. L'hypothèse implicite que nous faisons est donc que les variations des profils spectraux sont des données suffisamment fiables pour obtenir une segmentation significative. Nous montrons à l'aide d'une visualisation directe que cette hypothèse est très raisonnable (voir la figure 5). Il existe en fait de multiples possibilités pour encoder de façon binaire les variations ; nous en testons deux : une très simple qui donne déjà de bons résultats et une un peu plus compliquée qui permet d'améliorer ces derniers.
2. Le deuxième outil est la *version généralisée de la distance de Hamming* introduite dans [9]. Cette généralisation exploite fortement le fait que les caractères des chaînes manipulées sont binaires. Tout l'intérêt de cette nouvelle version est qu'elle permet de tenir compte, contrairement à la distance de Hamming usuelle, de la position des bits qui diffèrent. Pour ce faire, en plus des opérations traditionnelles d'insertion et de suppression, les auteurs considèrent une opération de shift qui permet de rendre compte du fait que par exemple la chaîne 1010 est plus proche de la chaîne 1100 que ne l'est la chaîne 1001. Si l'on se souvient que les codes binaires proviennent de variations de profils spectraux, cet aspect est essentiel.
3. Nous décrivons ensuite un algorithme de classification non supervisée par partitionnement ou *clustering*. Cet algorithme utilise les codes binaires et la distance de Hamming généralisée. L'idée naturelle est procéder à un *clustering* de type *k-means*. Il est bien connu cependant que les données binaires se prêtent mal à ce type d'approche car les descentes de gradient utilisées communément pour la recherche des barycentres ne respectent pas le caractère binaire des données. Nous

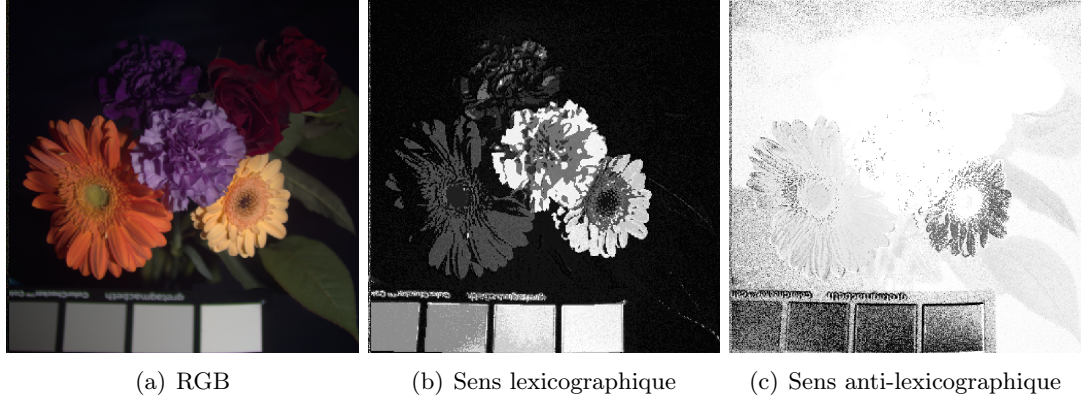


FIGURE 5 – Exemples de visualisation directe à l’aide de codages binaires.

sommes amenés à envisager une solution alternative basée sur l’utilisation de *k-modes* [31]. Nous discutons en particulier d’une méthode permettant d’optimiser le choix des modes initiaux de l’algorithme.

Il est à noter que les segmentations obtenues reflètent les variations de la forme des profils spectraux et non pas les variations des valeurs spectrales. Par ailleurs, là encore des extensions floues sont envisageables au niveau de l’algorithme de *k-modes*. Mentionnons également que cette approche soulève un certain nombre de questions sur lesquelles nous reviendrons, comme par exemple le problème du débruitage d’une chaîne binaire.

### 1.3.3 Segmentation riemannienne (voir chapitre 4, page 63)

Nous l’avons dit, peu de travaux sont consacrés à une approche purement géométrique de l’analyse du contenu d’images hyperspectrales. Une image hyperspectrale  $I$  peut être considérée<sup>4</sup> comme

1. une variété de dimension 2, i.e. une surface,  $S$  de  $\mathbb{R}^{\Lambda+2}$  paramétrée par

$$\varphi : (x, y) \longmapsto (x, y, I^1(x, y), \dots, I^\Lambda(x, y)) \quad (4)$$

où  $I^\lambda(x, y)$  désigne la valeur de la réflectance au pixel  $(x, y)$  pour la longueur d’onde  $\lambda$ ;

2. une famille  $(S_\lambda)$ , indicée par  $\lambda$ , de surfaces de  $\mathbb{R}^3$  paramétrées par

$$\varphi_\lambda : (x, y) \longmapsto (x, y, I^\lambda(x, y)) \quad (5)$$

3. une variété  $M$  dimension 3 de  $\mathbb{R}^4$  paramétrée par

$$\psi : (x, y, \lambda) \longmapsto (x, y, \lambda, I(x, y, \lambda) = I^\lambda(x, y)) \quad (6)$$

Le traitement et l’analyse géométriques des images sont basés en grande partie sur les propriétés métriques des graphes<sup>5</sup> tels que définis par (4), (5) ou (6). La géométrie

4. Il est traditionnel dans les approches géométriques du traitement des images de considérer dans les modèles théoriques que les données sont lisses, c’est-à-dire différentiables.

5. Ici au sens des variétés.

riemannienne fournit des outils puissants et opérationnels pour étudier ces propriétés [21, 24, 54]. L'un de ces outils est la *première forme fondamentale* ou encore *métrique riemannienne*. Il s'agit d'un tenseur d'ordre deux que l'on peut voir comme la donnée pour tout point  $(x, y)$  (resp. tout point  $(x, y, \lambda)$ ) d'une matrice  $2 \times 2$  (resp.  $3 \times 3$ ) symétrique définie positive. C'est à l'aide de ce tenseur que l'on peut mesurer la norme des vecteurs tangents à la surface et par suite la distance dite géodésique entre les points de cette même surface. Notre approche dans cette partie du travail repose sur les considérations suivantes.

1. Nous faisons l'hypothèse que les matrices  $2 \times 2$  ou  $3 \times 3$  symétriques définies positives, puisqu'elles mesurent les variations métriques des surfaces, sont des données pertinentes pour la segmentation. L'espace  $SPD(d)$  des matrices  $d \times d$  symétriques définies positives peut être lui-même muni d'une structure de variété riemannienne (voir [28] ou [73]) par le biais de la métrique dite de Fisher-Rao. Cette métrique, pour laquelle on dispose d'une expression explicite de la distance géodésique [53], est très pertinente pour la recherche de moyennes dans  $SPD(d)$  : ce sont les *moyennes de Karcher*. Nous reviendrons plus en détails sur l'ensemble de ces notions. Précisons simplement pour fixer les idées que si  $X$  et  $Y$  appartiennent à  $SPD(d)$ , alors la distance géodésique entre  $X$  et  $Y$  est donnée par la formule

$$d^2(X, Y) = \text{Trace}(\log(X^{-1}Y)^2) \quad (7)$$

où  $\log$  est le logarithme matriciel. La moyenne de Karcher  $\bar{X}$  de  $X_i$  matrices de  $SPD(d)$ , avec  $1 \leq i \leq n$ , est quant à elle définie par

$$\bar{X} = \underset{X \in SPD(d)}{\operatorname{argmin}} \sum_{i=1}^n d^2(X, X_i) \quad (8)$$

2. À chaque pixel d'une image hyperspectrale correspond soit une seule matrice  $G(x, y)$ , soit  $\Lambda$  matrices  $G_\lambda(x, y)$ ,  $1 \leq \lambda \leq \Lambda$ , de  $SPD(2)$ , soit une matrice  $G(x, y, \lambda)$  de  $SPD(3)$  suivant le point de vue que l'on adopte. Les espaces de caractéristiques correspondants sont  $SPD(2)$ ,  $SPD(2)^\Lambda$  et  $SPD(3)$ . Pour procéder à la segmentation, on utilise alors un *algorithme de k-means*.

Cette dernière approche tend à prendre en compte les variations spatiales locales de l'image ; elle n'est pas purement pixellique. Ceci est dû au fait que les matrices  $G(x, y)$ ,  $G_\lambda(x, y)$  ou  $G(x, y, \lambda)$  sont obtenues à partir des dérivées partielles de l'image. Les résultats obtenus montrent que la méthode classification est sensible au bruit, surtout dans  $SPD(2)$  et  $SPD(2)^\Lambda$ . Pour essayer de remédier à ce problème nous proposons d'utiliser un filtre médian défini à partir d'une notion de *médiane géométrique* dans l'espace  $SPD(d)$ . Nous reviendrons dans le chapitre 4 sur l'implémentation du calcul des médianes et des moyennes et en particulier sur notre choix d'utiliser un *algorithme récursif pour l'estimation des moyennes de Karcher* (voir la figure 6 et [30]). Précisons finalement que d'autres choix de distances sont possibles, comme la distance de Stein (voir [64]).

## 1.4 Implémentation et évaluation

Pour évaluer ces trois approches, nous avons utilisé les images hyperspectrales de deux bases de données [41] et [76]. Nous n'avons procédé à aucun pré ou post traitement, c'est la force



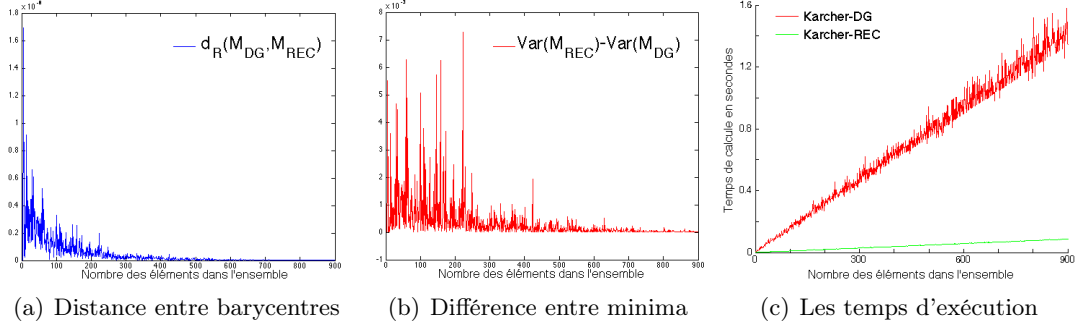


FIGURE 6 – Comparaison entre deux algorithmes de calcul de moyennes de Karcher (par descente de gradient, DG, et de façon réursive, REC).

de nos propositions. Les simulations du chapitre 4 (voir page 63) ont été réalisées en partie à l'aide du package [29] et de la toolbox [7].

Selon les images, l'évaluation consistera à comparer :

- visuellement la segmentation produite avec des segmentations produites par des méthodes de la littérature, ou avec la vérité-terrain (étiquetage supposé garanti exact des pixels)
  - numériquement la partition des pixels obtenue avec la vérité-terrain à l'aide d'une mesure de comparaison de partitions strictes (voir une revue complète dans [61]).
- Nous utiliserons le Rand Index [62] décrit à la fin de cette section.

#### 1.4.1 Algorithmes $k$ -means

En classification non supervisée, les méthodes de clustering ont pour but de partitionner un ensemble  $X = \{x_1, x_2, \dots, x_n\}$  de  $n$  objets décrits par  $p$  attributs en  $k$  classes appelées aussi *clusters*  $\omega_j$ ,  $j = 1, k$ . L'idée de base est que chaque objet doit être plus proche en termes de similarité des objets du cluster auquel il appartient que d'un quelconque objet d'un autre cluster. Un des algorithmes de classification non supervisée les plus couramment utilisés est l'algorithme des  $k$ -means [60]. À partir d'une initialisation de  $k$  centres (ou moyennes)  $C = \{c_1, \dots, c_k\}$  des clusters, il consiste à minimiser itérativement la distance intra-clusters :

$$J_d(C) = \sum_{j=1}^c \sum_{x \in \omega_j} d(x_i, c_j) \quad (9)$$

où  $d$  est une distance choisie par l'utilisateur, le plus souvent la distance euclidienne entre les objets et les centres :

$$d^2(x_i, c_j) = \|x_k - c_j\|^2. \quad (10)$$

Dans le chapitre 4 (page 63), la distance  $d$  sera la distance de Rao sur l'ensemble  $SPD(d)$  des matrices  $d \times d$  à coefficients réels symétriques définies positives. Notons que (9) est parfois donnée avec la distance au carré. Afin de faciliter le passage aux extensions de cet algorithme, on introduit souvent une matrice  $U$  ( $n \times k$ ) de partition stricte dont les éléments

$u_{ij}$  appartiennent à l'ensemble  $\in \{0, 1\}$ . Ainsi la fonctionnelle se réécrit :

$$J_d(U, C) = \sum_{i=1}^n \sum_{j=1}^k u_{ik} d(x_i, c_j) \quad (11)$$

où  $U$  est telle que :  $\forall 1 \leq i \leq n, \forall 1 \leq j \leq k$

$$\sum_{j=1}^k u_{ik} = 1, \quad (12)$$

$$0 < \sum_{i=1}^n u_{ik} < n. \quad (13)$$

La première contrainte traduit le fait que chaque objet n'appartient qu'à un seul cluster, la seconde qu'aucun des clusters n'est vide. L'algorithme est convergent, et atteint le plus souvent un minimum local de (11). L'optimisation  $k$ -means alterne deux phases d'actualisation, l'une pour les centres  $C$ , l'autre pour la partition  $U$  :

0. initialiser  $C^{(0)}$  (ou  $U^{(0)}$  et inverser les deux étapes d'actualisation), et le paramètre d'arrêt  $r^{(0)}$  à une valeur  $\varepsilon$  petite ;
1. calculer les distances entre les objets  $X$  et les centres  $C^{(t)}$  ;
2. actualiser  $U^{(t)}$  :  $\forall 1 \leq i \leq n, \forall 1 \leq j \leq k$ ,

$$u_{ij}^{(t)} = \begin{cases} 1 & \text{si } j = \operatorname{argmin}_{j=1,k} d(x_i, c_j^{(t)}) \\ 0 & \text{sinon} \end{cases} \quad (14)$$

3. actualiser  $C^{(t)}$  :  $\forall 1 \leq j \leq k$ ,

$$c_j^{(t)} = \frac{\sum_{i=1}^n u_{ij}^{(t)} x_i}{\sum_{i=1}^n u_{ij}^{(t)}} \quad (15)$$

4. tester la stabilité de la partition <sup>6</sup> :

$$r^{(t)} = \|U^{(t)} - U^{(t-1)}\| \quad (16)$$

aller en 1 tant que  $r^{(t)} \geq \varepsilon$ .

La plupart des algorithmes dérivés s'expriment de la même manière. Parmi les extensions, citons l'algorithme des  $k$ -means flous ([22, 5]) qui permet aux objets d'appartenir à plusieurs clusters. La fonctionnelle à minimiser devient :

$$J_{d,m}(U, C) = \sum_{i=1}^n \sum_{j=1}^k u_{ik}^m d^2(x_i, c_j) \quad (17)$$

où  $m > 1$  est un exposant de pondération (souvent fixé à 2), et les éléments de la matrice de partition floue (ou non stricte)  $u_{ij} \in [0, 1]$  représentent des degrés d'appartenance aux

6. Dans (16),  $\|\cdot\|$  désigne n'importe quelle norme matricielle, par exemple  $\|A\| = \max_i \max_j a_{ij}$ .



clusters. La matrice  $U$  est toujours soumise aux contraintes (12-13). La seule modification de l'algorithme porte sur les équations de mise-à-jour (14-15) :

$$u_{ij} = 1 / \sum_{j'=1}^k \left( \frac{d(x_i, c_j)}{d(x_i, c_{j'})} \right)^{\frac{2}{m-1}} \quad (18)$$

$$c_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m}. \quad (19)$$

Notons que formuler (17) avec la distance  $d$  au lieu de son carré  $d^2$  a pour seule conséquence d'enlever le carré dans (18).

Les algorithmes  $k$ -means (strict et flou) donnent de très bon résultats lorsque les attributs décrivant les objets sont sur une échelle non nominale car la minimisation est réalisée en modifiant les centres des clusters [33]. Par contre, on ne peut pas les utiliser directement sur des données catégorielles pour lesquels les attributs sont nominaux. Effectuer au préalable une transformation des attributs ne produit en général pas des résultats ayant du sens, même si les domaines sur lesquels sont définies les catégories sont ordonnés [57]. Il en va évidemment de même pour des données binaires qui ne sont qu'un cas particulier de données catégorielles. C'est pourquoi nous utiliserons au chapitre 3 (page 43) des variantes utilisant des modes fréquentiels.

#### 1.4.2 Rand Index

Soient deux partitions  $U$  et  $V$  d'un même ensemble de  $n$  objets en clusters éventuellement en nombres différents. Un indice de comparaison vise à mesurer à quel degré  $U$  et  $V$  s'accordent, c'est-à-dire à quel point les ensembles de clusters qui les composent sont compatibles. Deux partitions sont en accord total si et seulement si elles sont égales à une permutation des clusters près.

Le Rand Index [62] est fondé sur la construction d'une matrice ( $2 \times 2$ ) de contingence-paires<sup>7</sup>

$$M(U, V) = \begin{pmatrix} m_{11} & m_{10} \\ m_{01} & m_{00} \end{pmatrix} \quad (20)$$

dont les termes comptabilisent le nombre de paires d'objets qui :

- appartiennent conjointement au même cluster dans  $U$  et dans  $V$  (accord positif  $m_{11}$ ),
- appartiennent conjointement au même cluster dans  $U$  mais à des clusters différents dans  $V$  (désaccord  $m_{10}$ ),
- appartiennent conjointement à des clusters différents dans  $U$  mais au même cluster dans  $V$  (désaccord  $m_{01}$ ),
- n'appartiennent ni au même cluster dans  $U$  ni au même cluster dans  $V$  (accord négatif  $m_{00}$ ).

Il existe différentes approches (ensembliste, contingentielle, coïncidentielle) strictement équivalentes pour calculer ces termes (voir dans [61]) qui sont bien évidemment liés par :

$$m_{11} + m_{10} + m_{01} + m_{00} = \frac{n(n-1)}{2} = q, \quad (21)$$

---

7. Une paire est un ensemble de deux éléments distincts.

le nombre de paires que l'on peut former à partir d'un ensemble de  $n$  éléments. Le Rand Index, à valeurs dans  $[0,1]$ , est défini par :

$$RI(U, V) = \frac{m_{11} + m_{00}}{m_{11} + m_{10} + m_{01} + m_{00}} = \frac{m_{11} + m_{00}}{q} \quad (22)$$

et prend sa valeur maximale lorsque  $U$  et  $V$  sont en accord total puisque dans ce cas  $m_{11} = q$  et  $m_{10} = m_{01} = m_{00} = 0$ .

Notons que dans le cas de partitions strictes<sup>8</sup>, les termes de la matrice de contingence-paires sont respectivement le nombre de vrais positifs ( $m_{11}$ ), de faux positifs ( $m_{01}$ ), de faux négatifs ( $m_{10}$ ) et de vrais négatifs ( $m_{00}$ ), de sorte que le Rand Index devient une mesure d'exactitude, appelée *Overall Accuracy* (OA) dans la littérature.

---

8. Une partition stricte est composée de clusters mutuellement exclusifs, les objets ne peuvent appartenir qu'à un seul cluster.



## Chapitre 2

# Segmentation par similarité

---

Soit  $I$  une image hyperspectrale de taille spatiale  $M \times N$  et de taille spectrale  $\Lambda$ . À chaque pixel  $(m, n)$ ,  $1 \leq m \leq M$ ,  $1 \leq n \leq N$ , correspond une réponse spectrale  $I(m, n)$  constituée des valeurs de la réflectance aux différentes longueurs d'onde échantillonnées. De la même façon, à chaque longueur d'onde  $\lambda$ ,  $1 \leq \lambda \leq \Lambda$ , correspond une image en “niveau de gris”  $I(\lambda)$  de taille  $M \times N$ . Dans la suite<sup>1</sup>, les valeurs  $I(m, n, \lambda)$  de la réflectance correspondant aux pixels  $(m, n)$  et aux longueurs d'onde  $\lambda$  sont normalisées de telle sorte que  $I(m, n, \lambda)$  appartienne à l'intervalle  $[0, 1]$  pour tout  $(m, n)$  et tout  $\lambda$ . Notons

$E = \{1, 2, \dots, M\} \times \{1, 2, \dots, N\}$  et  $W = \{1, 2, \dots, \Lambda\}$ .

Dans ce chapitre, l'espace des caractéristiques est donc l'espace  $[0, 1]^\Lambda$  constitué des  $\Lambda$ -uplets de réels compris entre 0 et 1. La méthode de segmentation proposée consiste à appairer les pixels de l'image. Un appariement peut-être appréhendé selon différents points de vue. On peut considérer qu'il s'agit d'une relation de tolérance sur l'ensemble  $E \times E$ ; on peut également considérer qu'il s'agit d'un graphe non orienté dont l'ensemble des sommets est  $E$ . *In fine*, obtenir une segmentation à partir d'un appariement revient à considérer soit la plus petite relation d'équivalence qui contient la relation de tolérance définie par l'appariement, soit la fermeture transitive du graphe associé à l'appariement. Les objets de l'image sont alors soit les classes d'équivalence de la relation d'équivalence, soit obtenus en identifiant les sommets du graphe transitif qui sont reliés entre eux.

Le moyen le plus simple d'appairer les pixels de l'image est d'utiliser une mesure de similarité sur l'espace des caractéristiques  $[0, 1]^\Lambda$ . Il existe de nombreuses possibilités, voir par exemple [3]. Cependant, il faut tenir compte de deux contraintes : la similarité entre deux pixels doit se calculer rapidement et doit être de plus robuste au bruit.

Le coût algorithmique d'une telle méthode est important. Il apparaît clairement qu'une approche directe est déraisonnable. Pour pallier cette difficulté, il est préférable de travailler avec au moins deux “échelles”. De l'image initiale, on extrait des patches, ou imajettes, qui sont considérées comme des données locales de petites tailles. À cette échelle, la méthode décrite ci-dessus est opérante : les pixels sont appariés à l'aide d'une mesure de similarité spectrale; en résulte une segmentation de chacune des imajettes en un nombre raisonnablement petit d'objets. La même stratégie est appliquée à l'échelle globale avec cependant une différence notable qui permet de réduire le coût algorithmique. Pour obtenir la segmentation globale de l'image initiale, on procède à un appariement entre cette fois-ci

---

1. Cette normalisation ne concerne que ce chapitre.

les objets issus des segmentations locales. Là encore, nous utilisons une mesure de similarité spectrale mais maintenant entre objets, ou plus précisément entre classes d'équivalence locales. Nous évaluons les performances de cette approche en estimant le Rand Index entre les partitions obtenues et des partitions de vérités terrain.

## 2.1 Similarité spectrale entre pixels

### 2.1.1 Appariements

Nous introduisons dans ce paragraphe les définitions mathématiques des outils de base que nous utilisons dans la suite. Étant donnée une image hyperspectrale  $I$ , un appariement entre les éléments de l'ensemble  $E$  des pixels de  $I$  est une application

$$A : E \times E \longrightarrow \{0, 1\} \quad (23)$$

dont les propriétés minimales exigées sont :

$$A((m, n), (m, n)) = 1 \quad (24)$$

pour tout  $(m, n)$  et

$$A((m, n), (m', n')) = A((m', n'), (m, n)) \quad (25)$$

pour tous  $(m, n)$  et  $(m', n')$ .

Un tel appariement est une relation de tolérance, notée encore  $A$ , c'est-à-dire une relation binaire sur  $E \times E$  réflexive et symétrique à laquelle on peut associer un graphe non orienté, noté encore  $A$ , dont les sommets s'identifient aux éléments de l'ensemble  $E$  avec la propriété que deux tels sommets  $(m, n)$  et  $(m', n')$  sont reliés si et seulement si le pixel  $(m, n)$  est en relation avec le pixel  $(m', n')$ , ce que l'on notera  $(m, n)A(m', n')$ .

Une mesure de similarité [3] sur  $E$  est une application  $S : E \times E \longrightarrow [0, 1]$  vérifiant

$$S((m, n), (m, n)) = 1 \quad (26)$$

pour tout  $(m, n)$  et

$$S((m, n), (m', n')) = S((m', n'), (m, n)) \quad (27)$$

pour tous  $(m, n)$  et  $(m', n')$ .

Pour le problème qui nous concerne, les valeurs de similarité  $S((m, n), (m', n'))$  sont obtenues à partir de similarités marginales, ou similarités par bande,  $S_\lambda((m, n), (m', n'))$  pour  $1 \leq \lambda \leq \Lambda$ .

Le moyen le plus simple pour mesurer ces similarités marginales consiste à utiliser les fonctions

$$S_\lambda((m, n), (m', n')) = 1 - |I(m, n, \lambda) - I(m', n', \lambda)| \quad (28)$$

pour  $1 \leq \lambda \leq \Lambda$ .

**Définition 1.** Soit  $I$  une image hyperspectrale, la mesure de similarité spectrale  $S : E \times E \longrightarrow [0, 1]$  de  $I$  est définie par

$$S((m, n), (m', n')) = \prod_{\lambda=1}^{\Lambda} (1 - |I(m, n, \lambda) - I(m', n', \lambda)|) = \prod_{\lambda=1}^{\Lambda} S_\lambda((m, n), (m', n')) \quad (29)$$

À partir de la mesure de similarité  $S$ , il est facile de définir un appariement entre pixels, ou de façon équivalente, une relation de tolérance sur  $E \times E$ .

**Définition 2.** Avec les notations introduites ci-dessus, l'appariement spectral  $A_\tau$  est défini par

$$\boxed{(m, n)A_\tau(m', n') \iff S((m, n), (m', n')) \geq \tau} \quad (30)$$

où  $\tau \in [0, 1]$  est un seuil fixé proche de 1.

Évidemment, plus  $\tau$  sera choisi grand, moins de paires de pixels seront appariées, et plus d'objets seront déduits. Inversement, si  $\tau$  est trop petit, tous les pixels seront spectralement appariés de sorte que l'image sera segmentée en un seul objet. La relation de tolérance  $A_\tau$  peut être complétée en une unique relation d'équivalence  $R_\tau$ , c'est-à-dire satisfaisant de plus la condition de transitivité :

$$(m, n)R_\tau(m', n') \text{ et } (m', n')R_\tau(m'', n'') \implies (m, n)R_\tau(m'', n'') \quad (31)$$

Le graphe<sup>2</sup> dans le produit  $E \times E$  de la relation  $R_\tau$  est l'intersection des graphes des relations d'équivalence qui contiennent (en tant que graphes) le graphe de la relation  $A_\tau$ . Algorithmiquement,  $R_\tau$  est obtenue en considérant la fermeture transitive du graphe non orienté  $A_\tau$ . On utilise pour cela les puissances logiques itérées de la matrice d'adjacence de  $A_\tau$  (voir l'appendice 2.5, page 38).

**Remarque 1.** La mise en application de cette démarche soulève deux difficultés majeures.

1. La mesure de similarité spectrale est sensible au bruit. Il convient donc d'affiner l'approche décrite ci-dessus pour gagner en robustesse (voir la section 2.2 suivante, page 27).
2. Les images hyperspectrales sont des données particulièrement volumineuses. Le calcul des classes d'équivalence de la relation  $R_\tau$  ne peut être entrepris tel quel pour des raisons évidentes de complexité. Pour pallier cette difficulté, nous procédons dans un premier temps à des segmentations locales, c'est-à-dire en considérant des sous-images (ou imageries) de l'image initiale de taille spatiale réduite. Dans un second temps, les résultats locaux sont fusionnés afin d'obtenir une segmentation globale de l'image initiale. Ce dernier point fait l'objet de la section 2.4, page 34.

### 2.1.2 Prise en compte du bruit

Comme mentionné ci-dessus, la mesure de similarité  $S$  telle que définie par la formule (29) est sensible aux valeurs extrêmes. Afin de gagner en robustesse, nous introduisons une pénalisation. Notons

$$\tau = \prod_{\lambda=1}^{\Lambda} \tau_\lambda \quad (32)$$

le seuil introduit dans la définition 2, où les  $\tau_\lambda \in [0, 1]$ ,  $1 \leq \lambda \leq \Lambda$ , sont proches de 1 et éventuellement égaux.

---

2. Attention à ne pas confondre les deux sens du mot graphe.

Si  $((m, n), (m', n'))$  est un couple de pixels de  $I$ , soit  $\sigma_{((m, n), (m', n'))} \in \mathfrak{S}(\Lambda)$  la permutation telle que

$$\begin{aligned} S_{\sigma_{((m, n), (m', n'))}(1)}((m, n), (m', n')) &\leq S_{\sigma_{((m, n), (m', n'))}(2)}((m, n), (m', n')) \leq \dots \\ &\dots \leq S_{\sigma_{((m, n), (m', n'))}(\Lambda)}((m, n), (m', n')) \end{aligned} \quad (33)$$

Pour alléger les notations, posons

$$S^{(\lambda)}((m, n), (m', n')) := S_{\sigma_{((m, n), (m', n'))}(\lambda)}((m, n), (m', n')) \quad (34)$$

Notons également  $\tau^{(\lambda)} := \tau_{\sigma_{((m, n), (m', n'))}(\lambda)}$ .

**Définition 3.** Soit  $\eta$  un entier avec  $0 \leq \eta \leq \Lambda/2$  et soit

$$w_\eta((m, n), (m', n')) = \prod_{\eta < t \leq \Lambda - \eta} S^{(t)}((m, n), (m', n')) \quad (35)$$

$$w_\eta(\tau) = \prod_{\eta < t \leq \Lambda - \eta} \tau^{(t)} \quad (36)$$

La pénalisation  $P_\eta$  est définie par

$$P_\eta((m, n), (m', n')) = \begin{cases} -1 & \text{si } A_\tau((m, n), (m', n')) = 1 \text{ et } \exists \delta \leq \eta, w_\delta((m, n), (m', n')) \leq w_\delta(\tau) \\ 1 & \text{si } A_\tau((m, n), (m', n')) = 0 \text{ et } \exists \delta \leq \eta, w_\delta((m, n), (m', n')) \geq w_\delta(\tau) \\ 0 & \text{sinon} \end{cases} \quad (37)$$

On définit alors un nouvel appariement pénalisé (ou une nouvelle relation de tolérance pénalisée).

**Définition 4.** Avec les notations introduites ci-dessus, l'appariement spectral pénalisé  $A_{\tau, \eta}$  est défini par

$$\boxed{A_{\tau, \eta}((m, n), (m', n')) = A_\tau((m, n), (m', n')) + P_\eta((m, n), (m', n'))} \quad (38)$$

Traitons un exemple simple qui montre l'influence de la pénalisation sur l'appariement.

**Exemple 1.** Pour  $\Lambda = 5$  choisissons  $\tau_\lambda = 0,9$  avec  $1 \leq \lambda \leq 5$  et  $\eta = 1$ . Si

$$\begin{aligned} S^{(1)}((m, n), (m', n')) &= 0,2 \\ S^{(\lambda)}((m, n), (m', n')) &= 0,91 \text{ pour } 2 \leq \lambda \leq 5 \end{aligned}$$

alors

$$A_\tau((m, n), (m', n')) = 0 \text{ et } A_{\tau, \eta}((m, n), (m', n')) = 1 \quad (39)$$

De même, si

$$\begin{aligned} S^{(5)}((m, n), (m', n')) &= 1 \text{ et } S^{(4)}((m, n), (m', n')) = 0,9 \\ S^{(\lambda)}((m, n), (m', n')) &= 0,87 \text{ pour } 1 \leq \lambda \leq 3 \end{aligned}$$

alors

$$A_\tau((m, n), (m', n')) = 1 \text{ et } A_{\tau, \eta}((m, n), (m', n')) = 0 \quad (40)$$

De même que précédemment l'appariement  $A_{\tau,\eta}$  définit une relation d'équivalence  $R_{\tau,\eta}$ . Les classes d'équivalence de la relation  $R_{\tau,\eta}$  partitionnent <sup>a</sup> l'ensemble  $E$  des pixels de l'image hyperspectrale  $I$ . Chaque classe d'équivalence correspond à un "objet" de l'image.

a. C'est précisément ce dernier point qui justifie la fermeture transitive.

## 2.2 Segmentations locales

Nous nous intéressons ici à la segmentation d'images hyperspectrales de taille spatiale réduite, typiquement  $60 \times 60$ . De telles images sont considérées comme des patches locaux d'images hyperspectrales de taille spatiale plus importante. Dans la section suivante, nous expliquons comment agréger les segmentations locales pour obtenir une segmentation globale. Pour obtenir les segmentations locales, nous mettons en œuvre l'approche décrite dans les deux sections précédentes. Il s'agit donc d'apparier spectralement les pixels en tenant compte de la pénalisation.

### 2.2.1 Un exemple synthétique

Nous traitons pour commencer le cas d'une image multispectrale synthétique de taille  $60 \times 60 \times 14$ . La vérité-terrain de cette image consiste en 4 objets : les trois lettres "m", "i" et "a", plus le fond. Chaque image  $I(\lambda)$ ,  $1 \leq \lambda \leq 14$ , est dégradée par l'ajout d'un bruit gaussien additif  $\mathcal{N}(0, \sigma)$  de sorte de tester la robustesse de la méthode proposée. La figure 7 montre les images dégradées, bande par bande, pour  $\sigma = 0, 1$ . L'évaluation de la segmentation se fait à l'aide du Rand Index  $RI(S, G)$  (voir définition à la section 1.4, page 17) où  $S$  est la partition des pixels correspondant à la relation d'équivalence  $R_{\tau,\eta}$  et  $G$  est la partition vérité-terrain. La valeur 1 de  $RI(S, G)$  indique que les deux partitions coïncident parfaitement.

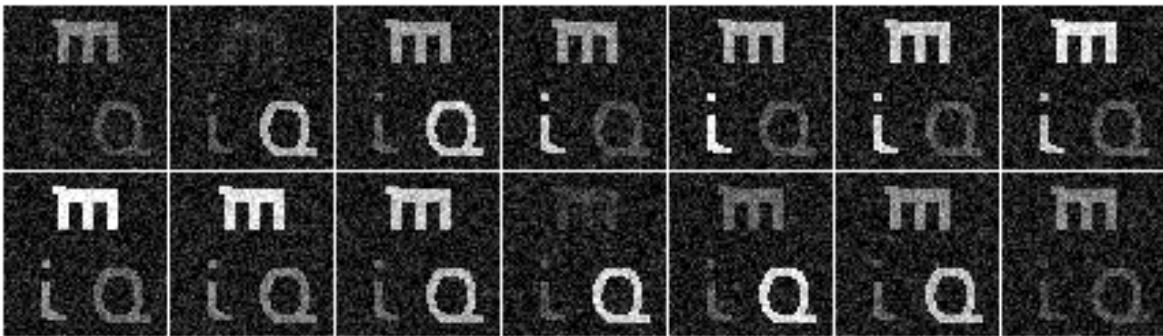


FIGURE 7 – Les  $\Lambda = 14$  bandes (de bas en haut et de gauche à droite) d'une imagerie synthétique bruitée avec  $\sigma = 0, 1$ .

Dans un premier temps, nous fixons  $\eta = 4$  et  $\tau_\lambda = (1 - \varepsilon)$  pour  $1 \leq \lambda \leq 14$ . Les tests sont réalisés avec 3 valeurs différentes de  $\sigma$  :  $\sigma = 0, 02$ ,  $\sigma = 0, 1$  et  $\sigma = 0, 4$  et en faisant varier  $\varepsilon$  entre 0 et 0,16. La courbe du Rand Index correspondante est représentée sur la figure 8 et quelques segmentations locales obtenues sur la figure 9 (celles repérées par le symbole  $\bullet$ ). On constate logiquement que la plage de variation de  $\varepsilon$  pour laquelle la segmentation est



parfaite diminue avec le bruit mais reste assez large même pour un bruit élevé :  $RI(S, G) = 1$  pour  $\varepsilon \in [0.042, 0.124]$  avec  $\sigma = 0,4$ . Lorsque  $\varepsilon$  est choisi trop grand, par exemple ici  $\varepsilon > 0,13$ , le seuil de similarité  $\tau$  (voir définition 2) atteint pour cette image une limite inférieure en deçà de laquelle tous les pixels sont spectrelement équivalents, induisant une valeur limite de  $RI(S, G) = 0,75$  qui correspond au ratio du nombre de pixels du fond sur le nombre total de pixels.

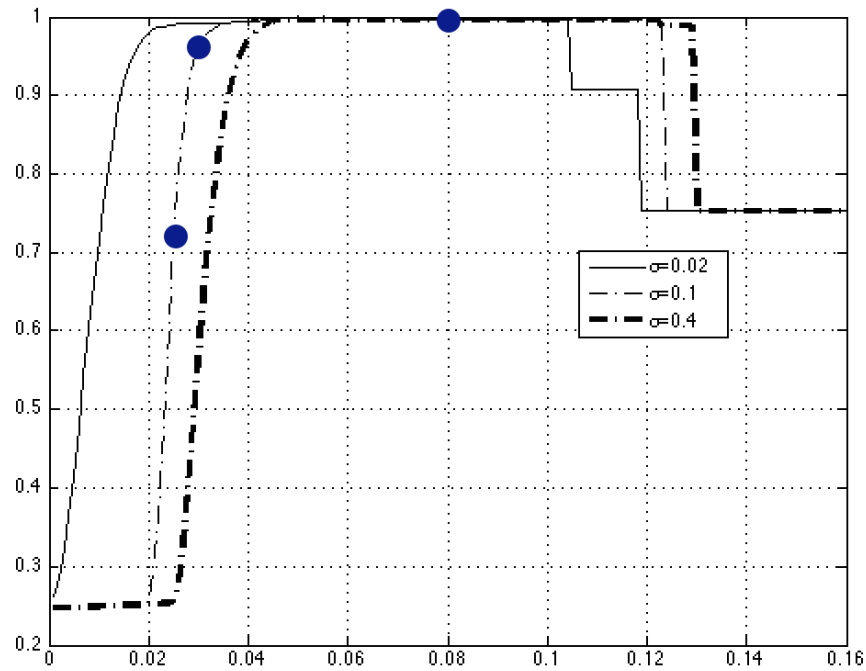


FIGURE 8 – Évaluation des résultats de segmentation locale pour l'imagette synthétique : la courbe du Rand Index en fonction de  $\varepsilon$  pour différentes valeurs de  $\sigma$ .

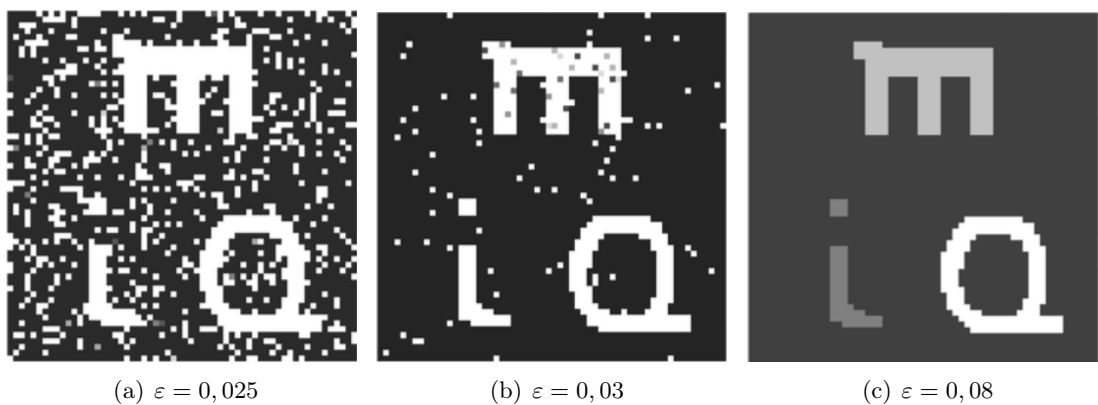


FIGURE 9 – Segmentation locale de l'imagette synthétique pour différentes valeurs de  $\varepsilon$ , avec  $\eta = 4$  et  $\sigma = 0,1$ .

Nous fixons maintenant les paramètres  $\sigma = 0,1$  et  $\varepsilon = 0,08$  pour évaluer l'influence du

paramètre de robustesse  $\eta$ . Les résultats obtenus pour  $\eta = 0$ ,  $\eta = 2$  et  $\eta = 4$  sont représentés à la figure 10; ils illustrent que la procédure de pénalisation permet de traiter efficacement les erreurs de segmentation dûes au bruit.

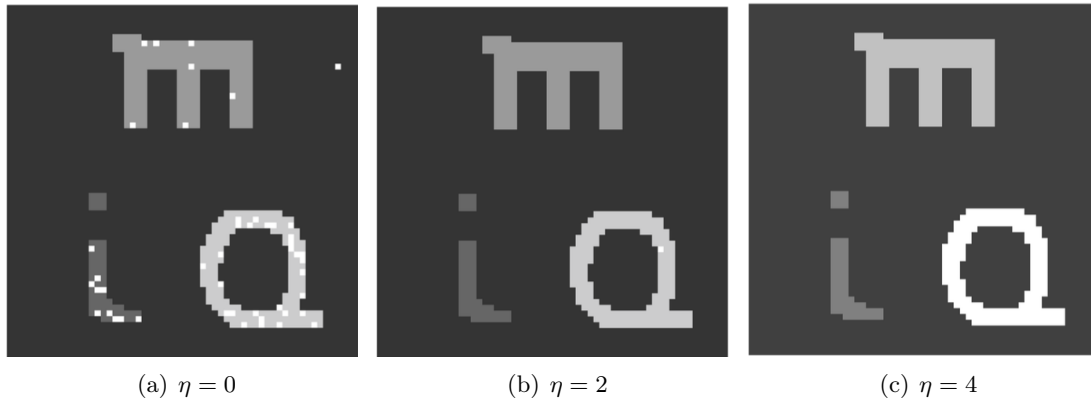


FIGURE 10 – Segmentation locale de l’imagette synthétique pour différentes valeurs du paramètre de robustesse  $\eta$ , avec  $\varepsilon = 0,08$  et  $\sigma = 0,1$ .

### 2.2.2 L’image “Pavia University Scene”

Nous traitons maintenant le cas d’une image de référence de la littérature. L’image initiale (voir [76]), “PaviaU”, est une image hyperspectrale à 103 bandes, de taille spatiale  $610 \times 340$ , dont on a extrait deux imagettes de taille spatiale  $60 \times 60$  notées  $I_1$  et  $I_2$  (voir la figure 11). La vérité terrain de l’image initiale fait apparaître 9 classes listées à la table 1, classes dont la présence dans les imagettes est indiquée en vert. Sur les images et imagettes les classes sont représentées du plus foncé au plus clair, le fond est noir (classe 0). Par exemple, sur la vérité-terrain de l’imagette  $I_2$  qui montre un échangeur autoroutier, on distingue :

- des arbres (classe 4) en clair,
- l’asphalte des routes (classe 1) en gris,
- le fond (classe 0) en noir.

L’usage d’une vérité-terrain pour évaluer un résultat de segmentation a ses limites, comme nous le montrons sur l’imagette  $I_2$ . En effet, les véhicules bien visibles sur  $I_2(30)$  ont été ignorés et affectés au fond.

Un exemple de segmentation de l’imagette  $I_1$  (respectivement  $I_2$ ) est donné à la figure 12(a) (respectivement 12(b)) obtenue avec pour paramètres  $\varepsilon = 0,0047$  et  $\eta = 30$  (respectivement  $\varepsilon = 0,0058$  et  $\eta = 16$ ). Le seuil  $\tau$  et le paramètre  $\eta$  sont plus grands pour l’imagette  $I_1$  que pour l’imagette  $I_2$ . En fait, il semble raisonnable de penser que plus il y a d’objets à segmenter, plus l’appariement doit être contraignant et les valeurs extrêmes écartées. Les Rand Index sont très élevés mais ne valent pas 1 aux raisons que la méthode de segmentation proposée a détecté des petits objets non référencés, en bas à droite pour  $I_1$  et en bas à gauche pour  $I_2$ . Il s’agit dans ce dernier cas du véhicule que l’on voit très distinctement sur  $I_2(30)$ .

**Remarque 2.** *D’une certaine façon, ceci montre que la méthode utilisée est à même de repérer de petites cibles anormalement présentes dans un environnement “cartographié”*

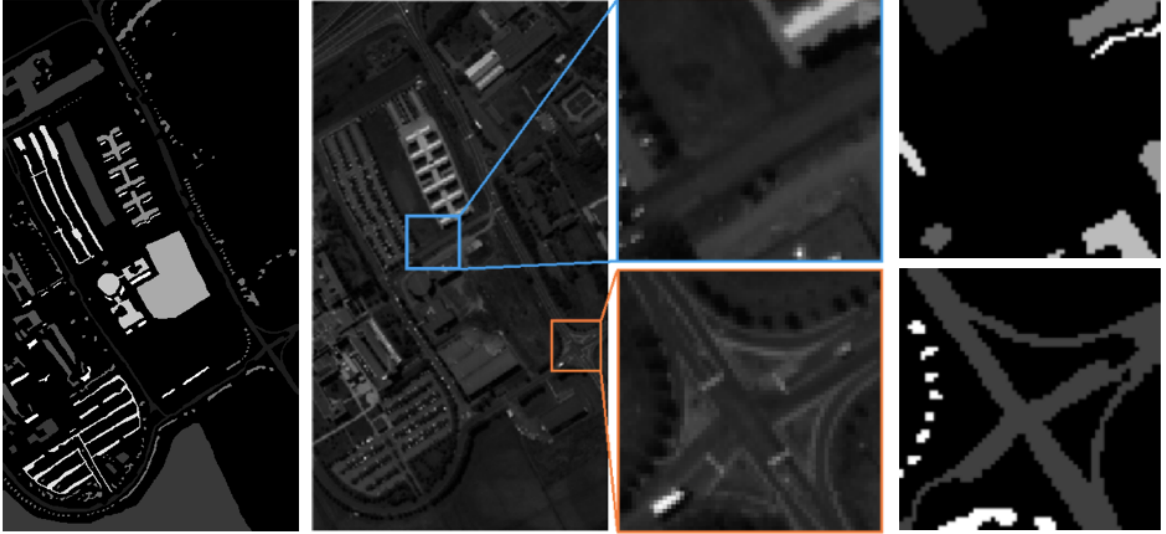


FIGURE 11 – PaviaU (de gauche à droite) : la vérité-terrain de l'image  $I$  ( $610 \times 340 \times 103$ ), l'image  $I(30)$ , deux imagettes  $I_1(30)$  (en haut) et  $I_2(30)$  (en bas), et leur vérité-terrain.

*spectralement. Ceci signifie précisément que l'algorithme de segmentation proposé est pertinent pour la détection de cibles.*

Comme dans toute méthode de segmentation, le choix du seuil (donc ici du paramètre  $\varepsilon$ ) influence de façon significative le résultat. À titre d'exemple, pour  $\varepsilon = 0,0051$  avec  $\eta = 30$ , le Rand Index pour l'imagette  $I_1$  devient 0,9833. La figure 13 montre en particulier que l'objet en bas à gauche de l'image n'est plus détecté et que les deux objets en haut à droite de l'image sont maintenant considérés comme appartenant à la même classe.

### 2.2.3 Sélection de bandes

Il peut être intéressant dans divers contextes d'identifier le domaine de réponse spectrale des objets détectés par l'algorithme de segmentation. Rappelons qu'un tel objet est déterminé par une classe d'équivalence  $\overline{(i, j)}$  pour la relation  $R_{\tau, \eta}$  où  $(i, j)$  est un pixel de l'objet. Pour

#	Classe (matériau)	nombre de pixels	$I_1$	$I_2$
1	Asphalte	6 631		
2	Herbes	18 649		
3	Gravier	2 099		
4	Arbres	3 064		
5	Tôles peintes	1 345		
6	Sol nu	5 029		
7	Bitume	1 330		
8	Briques auto-blocantes	3 682		
9	Ombres	947		

TABLE 1 – PaviaU : les classes de la vérité-terrain.

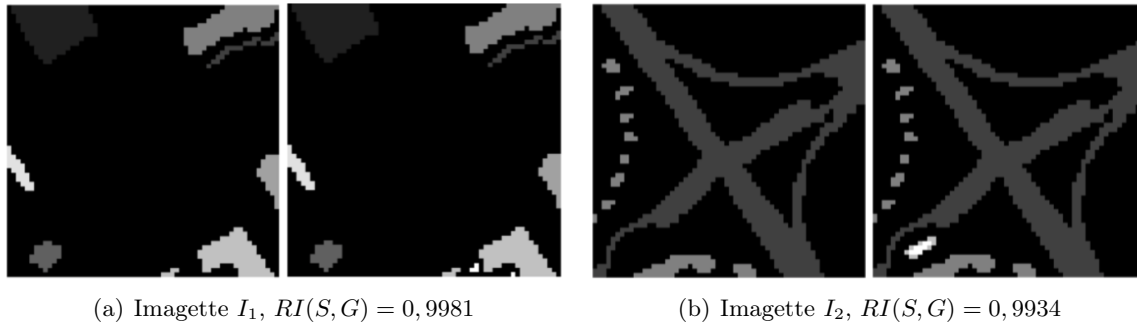


FIGURE 12 – Évaluation de la segmentation locale pour les deux imagettes test de PaviaU : à gauche les classes de la vérité terrain, à droite le résultat de la segmentation.



FIGURE 13 – Segmentation de l’imagette test  $I_1$  de PaviaU avec  $\eta = 30$  et  $\varepsilon = 0,0051$  (Rand Index=0,9833).

ce faire, on commence par calculer la moyenne arithmétique des réponses spectrales des pixels équivalents à  $(i, j)$  :

$$\langle \overline{(i, j)} \rangle_\lambda = \frac{1}{\#(i, j)} \sum_{(k,l) \in R_{\tau, \eta}(i, j)} I(k, l, \lambda) \quad (41)$$

et ce pour tout  $1 \leq \lambda \leq \Lambda$  et où  $\#(i, j)$  désigne le cardinal de la classe  $\overline{(i, j)}$ . Soit alors la plus grande de ces moyennes, prise comme réponse spectrale de l’objet identifié par une classe d’équivalence :

$$m[\overline{(i, j)}] = \max_{1 \leq \lambda \leq \Lambda} \langle \overline{(i, j)} \rangle_\lambda. \quad (42)$$

La table 2 montre les valeurs obtenues pour l’image l’image synthétique de la figure 7 (donc bruitée), en prenant les mêmes paramètres de segmentation qu’aux figures 9 et 10, à savoir  $\varepsilon = 0.08$  et  $\eta = 4$ . Notons que le fond de l’image “bg”, traité comme un objet, répond faiblement.

Un objet répond significativement dans la bande spectrale  $\lambda$  si

$$\langle \overline{(i, j)} \rangle_\lambda \geq \mu m[\overline{(i, j)}] \quad (43)$$

où  $\mu$  est un poids compris entre 0 et 1. La table 3 montre les bandes sélectionnées (en vert) avec  $\mu = 0,6$ . Les images spectrales sont redonnées aux fins d’illustration. Il est à noter que l’objet “a” répond significativement dans deux sous-ensembles  $\{2, 3\}$  et  $\{10, 11, 12, 13\}$

classe $(i, j)$	$m[(i, j)]$
"i"	0.94
"m"	0.96
"a"	0.88
"bg"	0.11

TABLE 2 – Maxima des moyennes des réponses spectrales des objets de l’imagerie synthétique.

disjoints de bandes spectrales, alors que l’objet “bg” (fond) est bien présent sur l’ensemble complet des bandes malgré ses réponses spectrales faibles.

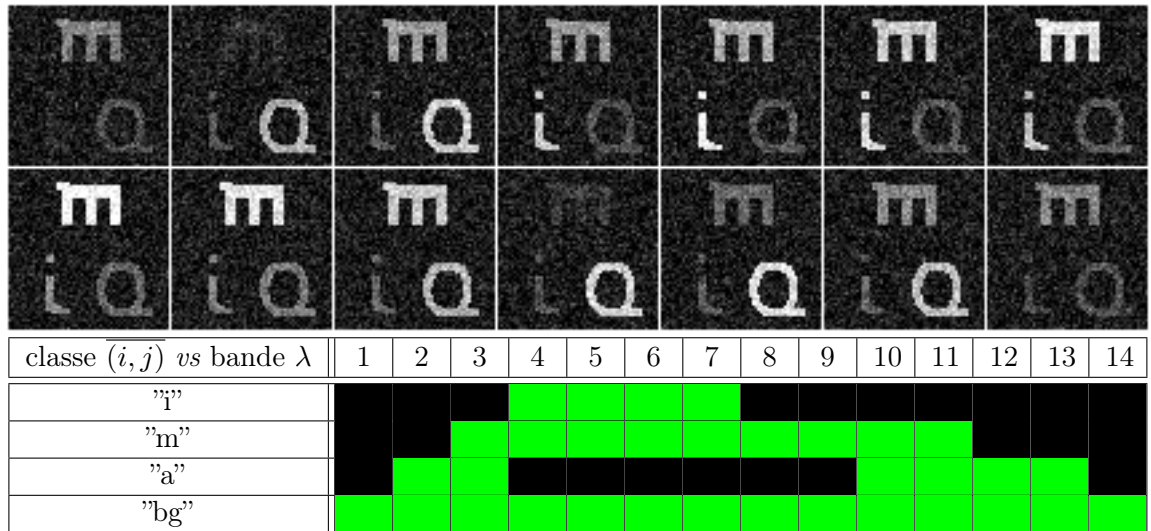
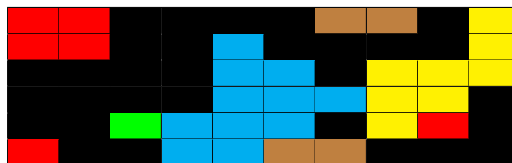


TABLE 3 – Les bandes spectrales significatives des objets de l’imagerie synthétique.

### 2.3 Similarité spectrale entre objets

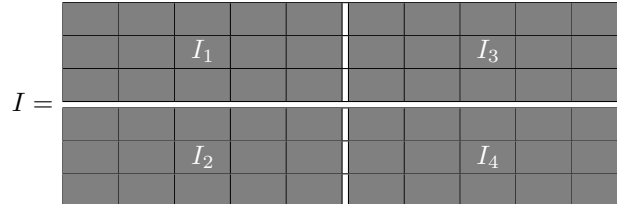
Comme mentionné plus haut, la méthode décrite ci-dessus est coûteuse en mémoire et en temps de calcul compte tenu du volume important des données hyperspectrales. C’est pourquoi nous n’avons traité jusqu’à présent que des imagerie de taille spatiale réduite (typiquement  $60 \times 60$ ). Le but de cette section est de présenter un algorithme qui permet de fusionner les résultats des segmentations locales (réalisées à partir d’imagerie) pour obtenir une segmentation globale des objets présents dans une image hyperspectrale du type “PaviaU” ( $340 \times 610$ ). Nous utilisons une nouvelle fois une mesure de similarité mais cette fois-ci entre objets.

Pour illustrer la démarche, nous considérons un cas simple, un “toy example”, correspondant à une image initiale de taille  $6 \times 10 \times 5$ , dont la vérité terrain serait :



Insistons ici sur le fait que notre but est de traiter des images de grande taille et que nous choisissons cet exemple de petite taille à des fins de compréhension.

1. Le découpage spatial de l'image. L'image initiale de taille  $M = 6$ ,  $N = 10$ ,  $\Lambda = 5$  est découpée en  $k = l \times c = 4$  ( $l = 2$  et  $k = 2$ ) imagettes  $I_i$ ,  $1 \leq i \leq k$ , de taille  $m = 3$ ,  $n = 5$ ,  $\Lambda = 5$  :



2. Les segmentations locales. L'algorithme de segmentation de la section 2.2 est appliqué à chaque imagerie  $I_i$  pour  $1 \leq i \leq k$ . Les valeurs des paramètres  $\tau$  et  $\eta$  sont identiques. Les objets de chaque image  $I_i$  sont numérotés de  $p_i$  à  $p_i + q_i$  où  $q_i + 1$  est le nombre d'objets détectés, y compris éventuellement le fond, avec l'ordre suivant : le nombre de pixels de l'objet  $p_i + j$  est supérieur ou égal au nombre de pixels de l'objet  $p_i + j + 1$  pour  $0 \leq j \leq q_i - 1$ . La numérotation globale se fait en imposant  $p_1 = 1$  et  $p_{i+1} = p_i + q_i + 1$ , l'indice  $i$  variant de 1 à  $k - 1$ . On note  $o = p_k + q_k$  et  $O = \{1, 2, \dots, o\}$ .

Sur notre exemple, on suppose que les  $k = 4$  imagettes ont été segmentées respectivement en 3, 4, 4 et 5 objets, pour un total de  $o = 16$  que certains chevauchent plusieurs imagettes. Dans la suite  $S = (S_{i,j})$  désigne la matrice de taille  $M \times N$  ainsi obtenue.

$S =$

	2	2	1	1	1	8	10	10	8	9
	2	2	1	1	3	8	8	8	8	9
	1	1	1	1	3	11	8	9	9	9
	4	4	4	4	5	13	13	14	14	12
	4	4	6	5	5	13	12	14	16	12
	7	4	4	5	5	15	15	12	12	12

3. La fusion des segmentations locales. On reprend l'approche de la section 1. L'idée est de définir un appariement entre les  $o$  objets obtenus par les segmentations locales et d'étendre cet appariement en une relation d'équivalence.

Remarque fondamentale. Pour les exemples réels, le nombre d'objets  $o$  est très petit comparé aux nombres de pixels de l'image initiale. La fermeture transitive nécessaire pour obtenir la relation d'équivalence est peu coûteuse.

L'information de départ dont on a besoin pour construire l'appariement est l'ensemble des valeurs médianes des réponses spectrales des pixels qui constituent chaque objet. Plus précisément, on définit pour  $1 \leq \lambda \leq \Lambda$  et  $1 \leq r \leq o$  (ici  $\Lambda = 5$  et  $o = 16$ )

$$\widetilde{M}(\lambda, r) = \text{median}\{I(i, j, \lambda), S(i, j) = r\} \quad (44)$$

pour obtenir une matrice  $\widetilde{M}$  de taille  $\Lambda \times o$ . On suppose pour illustrer la démarche que  $\widetilde{M}$  est donnée par

0.02	0.96	0.31	0.025	0.32	0.27	0.92	0.03	0.73	0.53	0.31	0.02	0.33	0.71	0.51	0.98
0.12	0.86	0.46	0.125	0.47	0.42	0.82	0.13	0.58	0.33	0.46	0.12	0.48	0.56	0.31	0.88
0.17	0.71	0.56	0.175	0.57	0.52	0.66	0.18	0.48	0.28	0.56	0.17	0.58	0.46	0.26	0.73
0.21	0.75	0.71	0.215	0.72	0.67	0.72	0.22	0.63	0.38	0.71	0.21	0.73	0.61	0.36	0.78
0.04	0.91	0.86	0.045	0.87	0.82	0.87	0.05	0.83	0.60	0.86	0.04	0.88	0.81	0.58	0.93

La mesure de similarité spectrale entre objets que nous considérons ici est définie comme suit.

**Définition 5.** La mesure de similarité spectrale  $\tilde{S} : O \times O \rightarrow [0, 1]$  entre objets de  $O$  est définie par

$$\tilde{S}(u, v) = \frac{1}{\Lambda} \sum_{t \in \Lambda} (1 - |\tilde{M}(\lambda, u) - \tilde{M}(\lambda, v)|) \quad (45)$$

Il s'agit maintenant de seuiller cette mesure de similarité spectrale en tenant compte des contraintes suivantes :

- (a) respect des segmentations locales de l'étape 1 ;
- (b) non appariement des objets dont les similarités spectrales sont trop basses ;
- (c) un objet d'une imagerie donnée doit être connecté à au plus un objet d'une imagerie différente de la première.

Tenant compte de toutes ces contraintes, nous proposons la définition suivante d'appariement spectral entre objets.

**Définition 6.** Soit  $\tau$  un seuil positif proche de 1. Avec les notations précédentes, l'appariement spectral  $\tilde{A}_\tau$  entre objets est défini par  $u\tilde{A}_\tau u$  pour tout  $u \in O$  avec de plus, pour tous  $1 \leq i \leq k$  et  $1 \leq j \leq q_i$  :

$$p_i(\text{non } \tilde{A}_\tau)(p_i + j) \text{ et } (p_i + j)(\text{non } \tilde{A}_\tau)p_i \quad (46)$$

et pour tous  $i_1 \neq i_2$ ,  $1 \leq i_1, i_2 \leq k$ ,  $0 \leq j_1 \leq q_{i_1}$  et  $0 \leq j_2 \leq q_{i_2}$  :

$$\begin{aligned} & (p_{i_1} + j_1)\tilde{A}_\tau(p_{i_2} + j_2) \iff (p_{i_2} + j_2)\tilde{A}_\tau(p_{i_1} + j_1) \iff \\ & \iff \tilde{S}(p_{i_1} + j_1, p_{i_2} + j_2) \geq \max_{0 \leq j \leq q_{j_2}} (\tau, \tilde{S}(p_{i_1} + j_1, p_{i_2} + j)) \end{aligned} \quad (47)$$

Étant donné un tel appariement spectral  $\tilde{A}_\tau$  entre objets, nous notons  $\tilde{R}_\tau$  la relation d'équivalence obtenue à partir de la fermeture transitive du graphe  $\tilde{A}_\tau$ .

## 2.4 Segmentation globale

### 2.4.1 Des segmentations locales à la segmentation globale

La condition (47) assure<sup>3</sup> en particulier qu'un objet d'une imagerie  $I_i$  est connecté à au plus un seul objet d'une imagerie  $I_j$  pour  $i$  différent de  $j$ . Voici ce que l'on obtient à partir de la matrice  $\tilde{M}$ . La matrice de similarité  $\tilde{S}$  est donnée par

3. D'où la présence des 0 marqués en rouge dans la matrice d'adjacence de  $\tilde{A}_\tau$  qui suit.

1	0.28	0.53	1.00	0.52	0.57	0.31	0.99	0.46	0.68	0.53	1.00	0.51	0.48	0.70	0.25
0.28	1	0.74	0.28	0.75	0.70	0.96	0.29	0.81	0.59	0.74	0.28	0.76	0.79	0.57	0.98
0.53	0.74	1	0.54	0.99	0.96	0.78	0.54	0.85	0.76	1.00	0.53	0.98	0.85	0.75	0.72
1.00	0.28	0.54	1	0.53	0.58	0.32	1.00	0.47	0.69	0.54	1.00	0.52	0.49	0.71	0.26
0.52	0.75	0.99	0.53	1	0.95	0.79	0.53	0.85	0.75	0.99	0.52	0.99	0.85	0.74	0.73
0.57	0.7	0.96	0.58	0.95	1	0.74	0.58	0.86	0.78	0.96	0.57	0.94	0.86	0.77	0.68
0.31	0.96	0.78	0.32	0.79	0.74	1	0.32	0.85	0.63	0.78	0.31	0.79	0.83	0.61	0.94
0.99	0.29	0.54	1.00	0.53	0.58	0.32	1	0.47	0.69	0.54	0.99	0.52	0.49	0.71	0.26
0.46	0.81	0.85	0.47	0.85	0.86	0.85	0.47	1	0.78	0.85	0.46	0.85	0.98	0.76	0.79
0.68	0.59	0.76	0.69	0.75	0.78	0.63	0.69	0.78	1	0.76	0.68	0.75	0.8	0.98	0.57
0.53	0.74	1.00	0.54	0.99	0.96	0.78	0.54	0.85	0.76	1	0.53	0.98	0.85	0.75	0.72
1.00	0.28	0.53	1.00	0.52	0.57	0.31	0.99	0.46	0.68	0.53	1	0.51	0.48	0.7	0.25
0.51	0.76	0.98	0.52	0.99	0.94	0.79	0.52	0.85	0.75	0.98	0.51	1	0.85	0.74	0.74
0.48	0.79	0.85	0.49	0.85	0.86	0.83	0.49	0.98	0.8	0.85	0.48	0.85	1	0.78	0.77
0.7	0.57	0.75	0.71	0.74	0.77	0.61	0.71	0.76	0.98	0.75	0.7	0.74	0.78	1	0.55
0.25	0.98	0.72	0.26	0.73	0.68	0.94	0.26	0.79	0.57	0.72	0.25	0.74	0.77	0.55	1

et la matrice d'incidence de l'appariement  $\tilde{A}_\tau$  par

1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Il ne reste plus qu'à appliquer l'algorithme de fermeture transitive pour fusionner les segmentations locales, c'est-à dire pour obtenir les objets présents dans l'image initiale. On obtient sur cet exemple la matrice  $\tilde{R}_\tau$  suivante

1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Au terme de ce processus, partant des segmentations locales suivantes :

2	2	1	1	1	8	10	10	8	9
2	2	1	1	3	8	8	8	8	9
1	1	1	1	3	11	8	9	9	9
4	4	4	4	5	13	13	14	14	12
4	4	6	5	5	13	12	14	16	12
7	4	4	5	5	15	15	12	12	12

on aboutit à la segmentation globale



2	2	1	1	1	1	6	6	1	5
2	2	1	1	3	1	1	1	1	5
1	1	1	1	3	3	1	5	5	5
1	1	1	1	3	3	3	5	5	1
1	1	4	3	3	3	1	5	2	1
2	1	1	3	3	6	6	1	1	1

en 6 classes d'objets, à comparer avec la vérité-terrain donnée à la page 33

### 2.4.2 L'image "Pavia University Scene"

Nous testons maintenant cette démarche sur notre image de référence. Rappelons qu'il s'agit d'une image  $340 \times 610 \times 103$ . Le choix des paramètres est le suivant et le même pour toutes les imagerie :  $\varepsilon = 0,0048$  et  $\eta = 30$ . La figure 14(a) montre le résultat des segmentations locales obtenues en suivant la procédure de la section 2.3, page 32. La figure 14(b) montre le résultat de la segmentation globale obtenue en suivant la procédure de la section 2.4, page 34. Pour cette dernière, le Rand Index de comparaison avec la vérité terrain est de 0,9934.

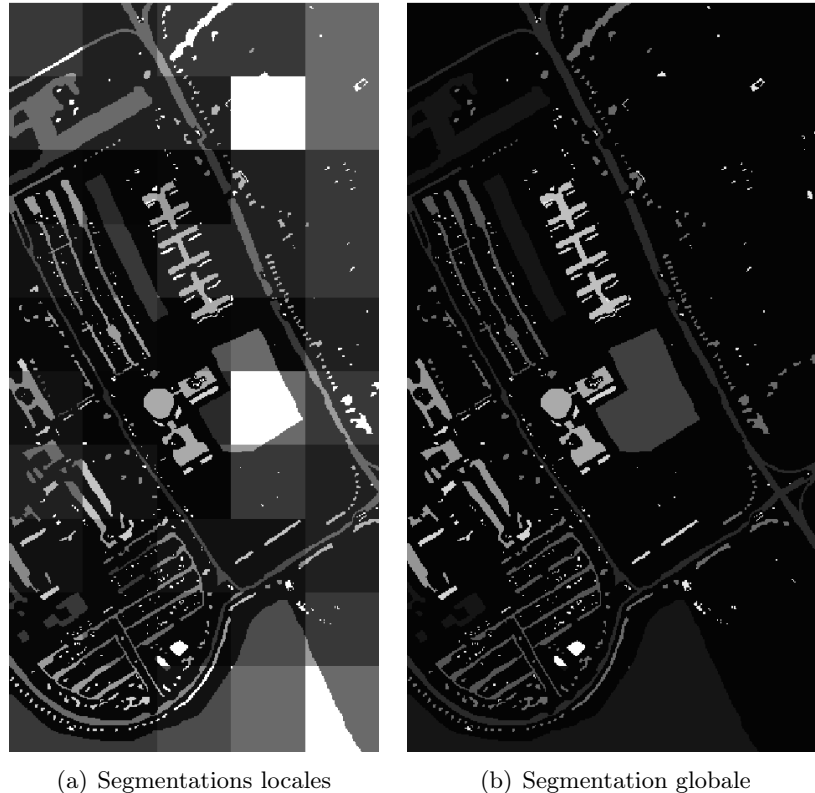


FIGURE 14 – Segmentations locales et globale de l'image PaviaU

Pour des raisons de procédure, les  $q_i$  objets détectés dans chaque imagerie  $i$  sont ordonnés de  $p_i = 1$  à  $p_i + q_i = 1 + q_i$ , car la numérotation globale ne permettrait pas une bonne visualisation. Un code couleur permettant une meilleure visualisation peut cependant être adopté afin de mieux rendre compte de la qualité des segmentations locale (voir figure 15) et globale (voir figure 16).

Les plages de bandes spectrales significatives sélectionnées selon la procédure décrite à la section 2.2.3 (page 30) à l'issue de la segmentation globale sont données à la table 4 pour deux valeurs de  $\mu$ . On constate tout d'abord que les plages sont logiquement moindres en nombre et en taille pour une valeur plus faible de  $\mu$ . Ensuite, et cela est dû à l'approche de segmentation locale proposée, comme déjà illustré aux figures 12(a) et 12(b), page 31, des objets supplémentaires de petite taille mais relativement nombreux sont détectés, certains correspondant à des objets bien visibles mais non référencés sur la vérité-terrain.

#	Classe (matériau)	Échantillons	Plages ( $\mu = 0.6$ )	Plages ( $\mu = \sqrt{2}/2$ )
0	Fond	161087	[70-103]	[73-103]
1	Asphalte	7117	[1-103]	[1-103]
2	Pré	18740	[72-103]	[74-103]
3	Gravier	2206	[1-103]	[17-103]
4	Arbres	3554	[76-103]	[77-103]
5	Tôles peintes	1475	[1-43] $\cup$ {103}	[4-40]
6	Sol nu	5072	[67-103]	[71-103]
7	Bitume	1400	[1-103]	[1-103]
8	Briques auto-blocantes	4043	[1-103]	[15-103]
9	Ombres	1062	[1-4] $\cup$ {101} $\cup$ {103}	[1-2]
10	Pré (2)	511	[72-103]	[75-103]
11	Supplément	1129	[1-103]	[4-103]
12	Bruit	4	[75-103]	[77-103]

TABLE 4 – Les plages de bandes spectrales significatives de l'image PaviaU

### 2.4.3 D'autres exemples

Nous illustrons la fiabilité de notre approche sur deux autres exemples issus de la même base d'images hyperspectrales ([76]) :

- L'image "Salinas" est de taille  $512 \times 217 \times 204$ . Le résultat de segmentation montré à la figure 17(b) est obtenu avec les paramètres  $\varepsilon = 0,041$  et  $\eta = 12$ ; le Rand Index de comparaison avec la vérité terrain est de 0,9948.
- L'image "Indiana" est de taille  $145 \times 145 \times 200$ . Le résultats de segmentation

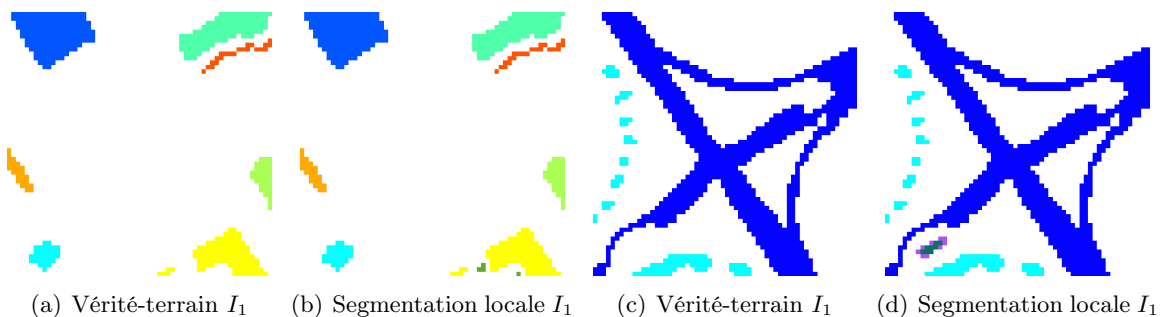


FIGURE 15 – Segmentation locale de l'image PaviaU

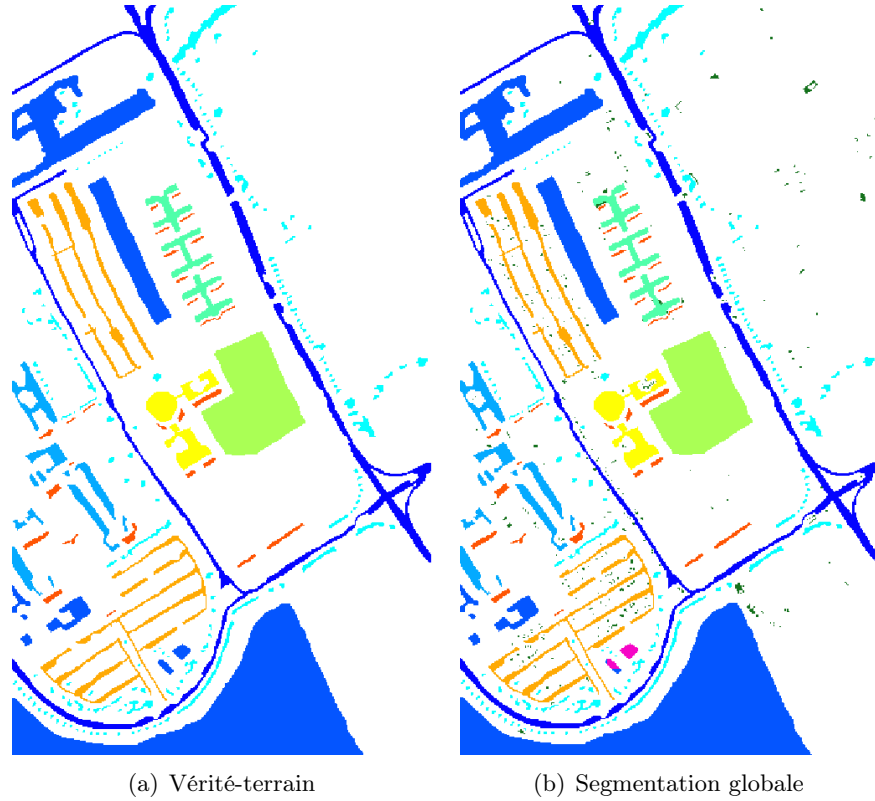


FIGURE 16 – Segmentation globale de l'image PaviaU

correspondant à la figure 18(b) sont obtenus avec les paramètres  $\varepsilon = 0,045$  et  $\eta = 7$  ; le Rand Index de comparaison avec la vérité terrain est de 0,9728.

Ces deux nouveaux exemples montrent effectivement que notre approche donne des résultats tout à fait satisfaisants.

## 2.5 Appendice

Nous donnons dans cet appendice une courte démonstration de l'algorithme de fermeture transitive utilisé pour obtenir les relations d'équivalence à partir des appariements. Soit  $A$  la matrice d'adjacence<sup>4</sup> d'une relation de tolérance  $A$ . Soit  $R$  la matrice d'adjacence du graphe non orienté qui est la fermeture transitive de  $A$  ( $R$  est la relation d'équivalence associée à  $A$ ). Nous notons  $N$  la taille de ces deux matrices<sup>5</sup>, i.e.  $A$  et  $R$  appartiennent à  $\mathcal{M}_N(\{0,1\})$ . Le produit matriciel utilisé ici est le produit logique (utilisant donc la multiplication et l'addition logiques). L'égalité  $A^N = R$  découle du lemme suivant.

**Lemme 1.** *Pour tous  $i$  et  $j$  appartenant à  $\{1, 2, \dots, N\}$ ,*

$$A^N(i, j) = 0 \iff R(i, j) = 0 \quad (48)$$

4. Nous adoptons les mêmes notations pour la relation de tolérance, l'appariement, le graphe ou la matrice d'adjacence de ce dernier.

5. Ici  $N$  n'est pas relié à une taille d'image.

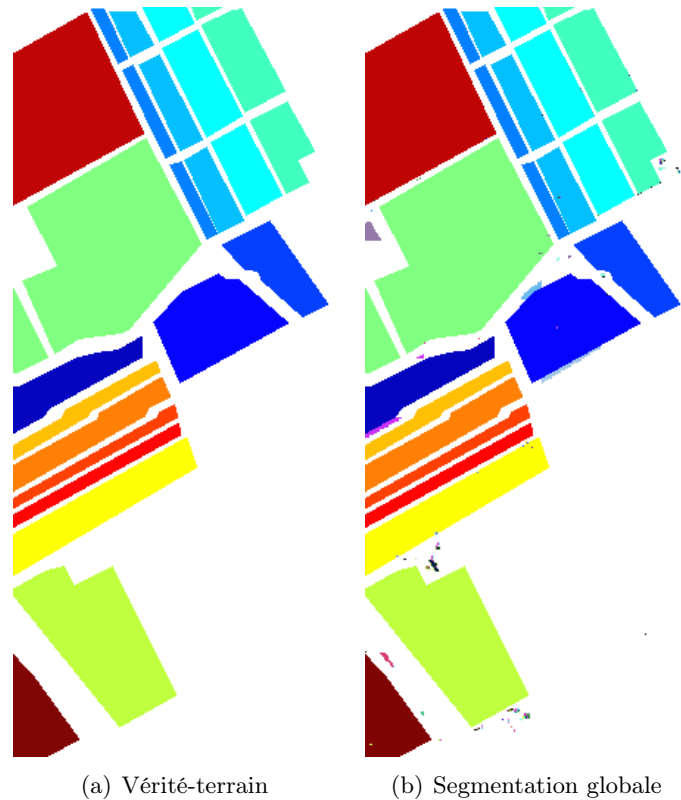


FIGURE 17 – Segmentation de l'image Salinas

**Preuve.** Nous commençons par vérifier que  $R(i, j) \neq 0$  implique  $A^N(i, j) \neq 0$ . Dire que  $R(i, j)$  est non nul revient à dire qu'il existe une suite d'entiers

$$n_0 = i, n_1, \dots, n_s, n_{s+1} = j \quad (49)$$

telle que

$$R(n_k, n_{k+1}) = 1 \quad (50)$$

pour tout  $0 \leq k \leq s$ .

On vérifie facilement les deux points suivants :

1.  $A^l(n_0, n_l)$  est non nul pour tout  $1 \leq l \leq s + 1$ . En particulier  $A^{s+1}(i, j) \neq 0$ .
2.  $A^{s+1}(i, j) \neq 0$  implique  $A^N(i, j) \neq 0$ .

ce qui permet de conclure.

Pour vérifier que  $A^N(i, j) \neq 0$  implique  $R(i, j) \neq 0$  on procède par récurrence sur  $l$  afin de montrer que : pour tout  $1 \leq l \leq N$ ,  $A^l(i, j) \neq 0$  implique  $R(i, j) \neq 0$ .

Il est bien connu que dans la plupart des cas il n'est pas nécessaire de calculer la puissance d'ordre  $N$  de la matrice  $A$ . Ceci est dû au fait que si  $A^l = A^{l+1}$  pour un certain  $1 \leq l < N$  alors  $A^l = A^N = R$ .

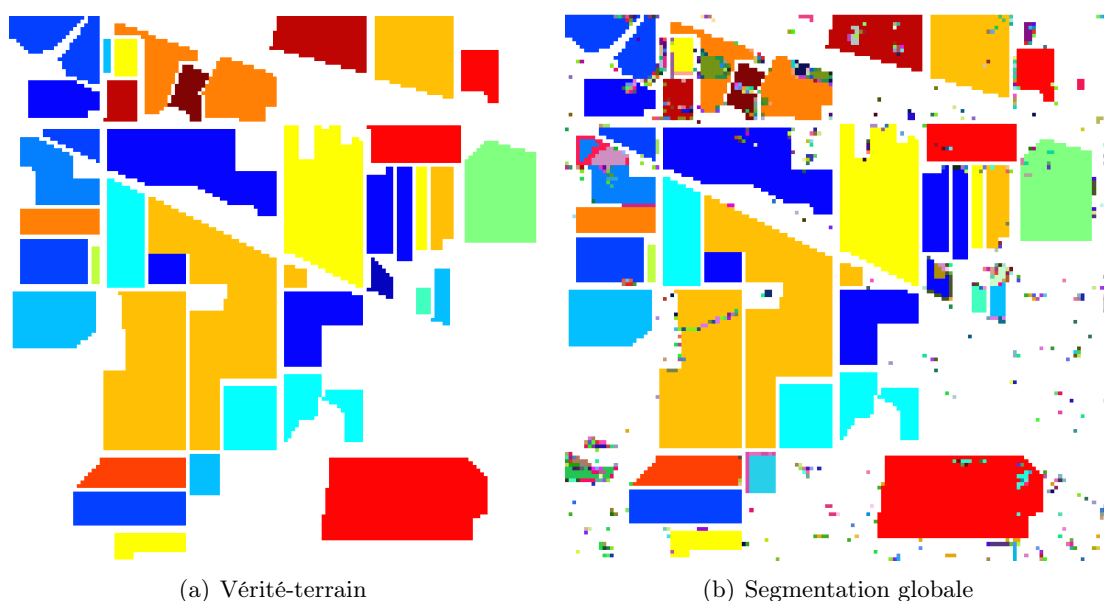


FIGURE 18 – Segmentation de l'image Indiana

## 2.6 Conclusions et perspectives du chapitre

Nous avons montré dans ce chapitre comment aborder le problème de la segmentation non supervisée d'images hyperspectrales en considérant l'espace des caractéristiques  $[0, 1]^A$  et suivant une approche de type "Graph Embedding". La méthode proposée utilise essentiellement deux mesures de similarité spectrale, l'une pour appairier les pixels des imagerie, l'autre pour fusionner les objets détectés dans les imagerie. Comme nous l'avons expliqué, ces deux étapes sont nécessaires pour remédier à la complexité des algorithmes mis en œuvre.

Les deux mesures de similarité sont des "paramètres" de la méthode. Nous nous sommes volontairement limités dans ce qui précède aux mesures des définitions 1 (voir page 24) et 5 (voir page 34). Ce sont les mesures les plus simples et les plus naturelles que l'on peut choisir. Il existe beaucoup d'autres choix et l'on peut facilement envisager d'étendre ces définitions à l'aide d'opérateurs flous tels que les  $t$ -normes et les  $t$ -conormes ou à l'aide de fonctions d'agrégation plus générales (voir par exemple [27, 38]).

Par ailleurs, il est également possible d'aborder le problème de la segmentation des images multispectrales de façon marginale en utilisant des opérateurs de similarité spatiale par blocs, comme dans [43], les résultats de similarité "par bande" étant dans un second temps agrégés. Nous n'avons pas approfondi cette piste malgré l'avantage qu'elle présente de rendre compte de la cohérence spatiale des images. Cette approche fait partie de nos perspectives de recherche future.

Mentionnons également que le découpage spatial de l'image initiale en imagerie que nous avons effectué peut être répété si besoin de façon pyramidale. Ainsi, le résultat de la segmentation globale de l'image synthétique (voir la fin de la section 2.4.1, page 36) pourrait être le cas échéant considéré après renumérotation des classes comme une segmentation locale à fusionner avec d'autres, comme illustré ci-dessous.

4	4	1	1	1	1	5	5	1	3									
4	4	1	1	2	1	1	1	1	3									
1	1	1	1	2	2	1	3	3	3	18								
1	1	1	1	2	2	2	3	3	1	14								
1	1	6	2	2	2	1	3	4	1	14								
4	1	1	2	2	5	5	1	1	1	14								
8	7	7	11							21								
8																		
8																		
7																		

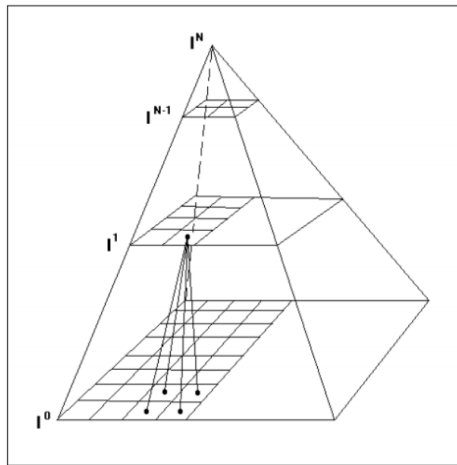


FIGURE 19 – Approche pyramidale de la segmentation



## Chapitre 3

# Segmentation par codage binaire

---

Compte tenu du mode d’acquisition, l’information exploitable associée à chaque pixel  $(m, n)$  d’une image hyperspectrale est le profil  $I(m, n)$  défini par

$$I(m, n) : \lambda \mapsto I(m, n, \lambda) \quad (51)$$

que l’on peut appréhender, surtout lorsque le nombre de bandes spectrales est grand, comme une courbe. Aborder le problème de la segmentation en choisissant comme espace de caractéristiques l’espace de telles courbes spectrales est un problème délicat qui nécessite en particulier de définir une distance pertinente entre courbes. Nous nous proposons ici de réduire significativement la complexité d’une telle approche en ne considérant que les variations des profils spectraux. L’hypothèse implicite que nous faisons donc dans la suite est que ces variations sont des caractéristiques suffisamment fiables pour notre étude. Les tests de visualisation que nous effectuons dans la section 3.1 montrent que cette hypothèse est tout-à-fait raisonnable. La démarche que l’on adopte s’inscrit dans la lignée des méthodes dites de “Spectral Hashing” ou “Semantic Hashing” (voir par exemple [75]) utilisées pour la recherche d’information. Elle comporte essentiellement trois étapes.

1. Les données sont converties en chaînes binaires à l’aide d’un encodage significatif.
2. Une distance entre chaînes binaires est définie de façon à refléter la distance entre les données.
3. Les données sont classifiées à l’aide d’un algorithme utilisant les chaînes binaires produites par l’encodage et la distance entre ces chaînes.

Il existe de multiples façons de convertir les variations spectrales en codes binaires. Nous commençons par un codage très naturel et très simple, voir la section 3.1 page suivante.

Nous traitons également d’un exemple un peu moins évident qui donne de meilleurs résultats notamment pour la détection d’anomalies (voir la section 3.3.3, page 55).

La distance classique entre chaînes de caractères est la distance de Hamming. Dans le contexte qui est le nôtre, cette distance n’est plus pertinente car elle ne prend pas en compte les positions où les caractères diffèrent. Par ailleurs, les chaînes que nous manipulons sont des chaînes binaires et non pas des chaînes de caractères plus généraux. Ceci nous permet d’utiliser la version généralisée de la distance de Hamming introduite dans [9] (voir la section 3.2, page 48). Mentionnons simplement ici que cette distance de Hamming généralisée fait intervenir, en plus des opérations classiques d’insertion et de suppression,



une opération de shift qui permet de rendre plus proches des chaînes binaires dont les bits qui diffèrent sont proches. Il est à noter que la seule information dont il est fait usage est la position des 1-bits dans les chaînes.

Il est bien connu que les algorithmes de  $k$ -means se prêtent mal à la classification de données binaires. En effet, les descentes de gradient couramment utilisées pour la recherche des moyennes (ou barycentres) sont des procédés de minimisation qui ne respectent pas le caractère binaire des données. Pour pallier cette difficulté, nous avons recours à un algorithme de classification de type  $k$ -modes dans l'esprit de [31]. Nous discutons également dans la section 3.3 (page 52) d'une stratégie pour le choix des modes d'initialisation.

Remarque fondamentale. Il convient enfin de noter que les segmentations obtenues tiennent compte des variations de la forme des profils spectraux et non des variations des valeurs spectrales.

### 3.1 Codage binaire de variations spectrales

Soit  $I$  une image hyperspectrale de taille spatiale  $M \times N$  et de taille spectrale  $\Lambda$ . Rappelons que  $E = \{1, 2, \dots, M\} \times \{1, 2, \dots, N\}$  et  $W = \{1, 2, \dots, \Lambda\}$ . Les variations du profil  $I(m, n)$  sont données par

$$\Delta I(m, n)(\lambda + 1) = I(m, n, \lambda + 1) - I(m, n, \lambda) \quad (52)$$

pour  $1 \leq \lambda \leq \Lambda - 1$ . D'une certaine façon, ces variations encodent "la forme" du profil spectral indépendamment des valeurs puisque

$$\Delta[I(m, n) + a](\lambda + 1) = \Delta I(m, n)(\lambda + 1) \quad (53)$$

pour toute constante  $a$ .

**Définition 7.** Le codage binaire  $I_b$  de l'image hyperspectrale  $I$  est l'application de  $E$  à valeurs dans  $\{0, 1\}^{\Lambda-1}$  définie par

$$I_b(m, n)(\lambda) = \begin{cases} 1 & \text{si } \Delta I(m, n)(\lambda + 1) \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (54)$$

pour  $1 \leq \lambda \leq \Lambda - 1$ .

À chaque pixel  $(m, n)$  de  $E$  on associe donc une suite de  $\Lambda - 1$  éléments de l'ensemble  $\{0, 1\}$ .

**Remarque 3.** Deux stratégies sont possibles pour exploiter ce codage binaire.

1. La première consiste à convertir chaque  $I_b(m, n)$  en un nombre décimal pour obtenir une nouvelle image, notée encore  $I_b$ , cette fois-ci en niveau de gris<sup>1</sup>. Cette dernière image peut-être traitée et analysée par les méthodes usuelles ([26]). Dans la section suivante, nous abordons sous cet angle le problème de la visualisation des données hyperspectrales en appliquant un algorithme de segmentation issu de la géométrie riemannienne. Il faut signaler que ce procédé qui consiste à traiter l'image  $I_b$  comme une image en niveau de gris après conversion des codes binaires en valeurs décimales donne plus de poids aux variations spectrales des premières longueurs d'onde qu'aux

---

1. L'image obtenue par ce procédé est codée "en  $\Lambda - 1$  bits".

variations spectrales des dernières longueurs d'onde. À titre d'exemple les codes binaires 11000010, 11000000 et 10000010 correspondent respectivement aux valeurs décimales 194, 192 et 130. Si l'on inverse le sens de lecture de ces chaînes, on obtient respectivement : 01000011 soit 67, 00000011 soit 3 et 01000001 soit 65. Pour pallier cet inconvénient, on peut "symétriser" la conversion décimale. Dans la suite, nous noterons  $I_b^+$  l'image en niveaux de gris obtenue en convertissant les codes binaires de façon usuelle, c'est-à-dire en les lisant de gauche à droite, et  $I_b^-$  l'image en niveaux de gris obtenue en lisant les codes binaires de droite à gauche. L'image  $I_b^-$  est en quelque sorte le "négatif" de l'image  $I_b^+$ .

2. La seconde stratégie consiste à garder le codage binaire pour l'exploiter en tant que tel. C'est ce point de vue que l'on adopte dans la section 3.3 (page 52) pour segmenter l'image initiale  $I$  par un algorithme de classification. Cet algorithme utilise une généralisation de la distance de Hamming qui permet de tenir compte de l'espacement entre les modifications de bits. Nous rappelons dans la section 3.2 (page 48) les principaux résultats issus de [9] concernant cette distance. Les données étant binaires, nous optons finalement pour une méthode de partitionnement de type  $k$ -modes dans la lignée de [31].

### 3.1.1 Visualisation

L'objectif principal de cette sous-section est de montrer que le codage binaire introduit précédemment est un outil simple qui permet de rendre compte efficacement du contenu d'images hyperspectrales. Pour les illustrations, nous utilisons les images suivantes :

- "Flowers" et "Balloons" de taille  $512 \times 512 \times 31$  ; les images RGB, la vérité-terrain, sont acquises avec un illuminant normalisé de lumière du jour (D65) ([41]).
- "PaviaU" de taille  $340 \times 610 \times 103$  ; outre la vérité-terrain, l'image de référence est l'image  $I(60)$  correspondant à la bande  $\lambda = 60$  ([76]).

### 3.1.2 Exemples de visualisation directe

Les figures 20 et 21 montrent ce que l'on obtient en affichant directement les images  $I_b^+$  et  $I_b^-$  obtenues après conversion décimale des codes binaires rendant compte des variations spectrales.

### 3.1.3 Segmentation riemannienne

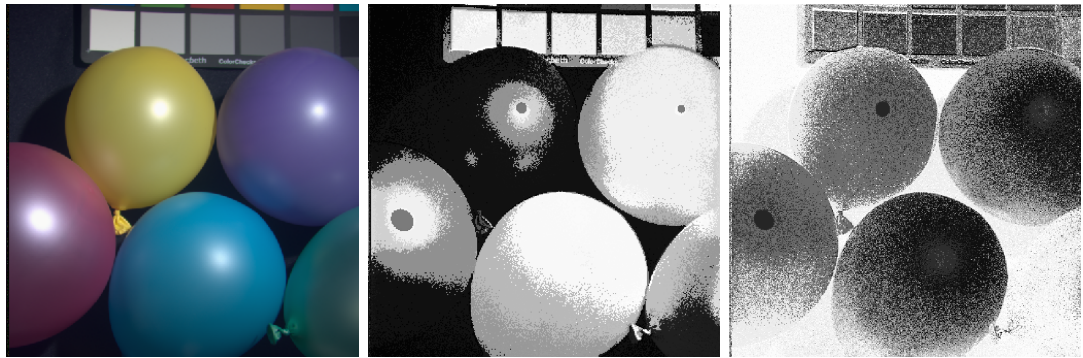
Nous décrivons ici une méthode de détection de contours basée sur les propriétés métriques du graphe de l'image  $I_b$ . Bien qu'elle soit plus ou moins classique, nous allons détailler les points importants (voir par exemple [4]).

Pour les besoins de la modélisation, les applications  $I_b^+$  et  $I_b^-$  sont vues comme des applications d'un domaine  $\Omega$  de  $\mathbb{R}^2$  à valeurs dans  $\mathbb{R}$  et supposées suffisamment lisses pour que les calculs qui suivent aient un sens. Le domaine  $\Omega$  est muni des coordonnées  $(x, y)$  et les réels  $I_b^+(x, y)$  et  $I_b^-(x, y)$  sont les niveaux de gris correspondant au code binaire associés au pixel  $(x, y)$  c'est-à-dire

$$I_b^+(x, y) \simeq [I_b(x, y)(1), I_b(x, y)(2), \dots, I_b(x, y)(\Lambda - 1)] \quad (55)$$



(a) Flowers.



(b) Balloons.

FIGURE 20 – Conversion décimale du codage binaire pour les images Flowers et Balloons : RGB (à gauche),  $I_b^+$  (au centre),  $I_b^-$  (à droite).

$$I_b^-(x, y) \simeq [I_b(x, y)(\Lambda - 1), I_b(x, y)(\Lambda - 2), \dots, I_b(x, y)(1)] \quad (56)$$

Nous ne traitons dans ce qui suit que le cas des images  $I_b^+$ , le cas des images  $I_b^-$  étant tout à fait analogue.

À l'image, ou l'application,  $I_b^+$  on, associe une variété  $S_b^+$  de dimension 2, c'est-à-dire une surface, paramétrée par la carte globale

$$\begin{aligned} \varphi_b^+ : \Omega &\longrightarrow \mathbb{R}^3 \\ \varphi_b^+(x, y) &= (x, y, I_b^+(x, y)) \end{aligned} \quad (57)$$

Autrement dit  $S_b^+$  est le graphe dans  $\mathbb{R}^3$  de l'application  $I_b^+$ . Soit  $p = \varphi_b^+(x, y)$  un point de  $S_b^+$ , alors les deux vecteurs

$$(\varphi_b^+)_x = \begin{pmatrix} 1 \\ 0 \\ (I_b^+)_x \end{pmatrix} \quad (\varphi_b^+)_y = \begin{pmatrix} 0 \\ 1 \\ (I_b^+)_y \end{pmatrix} \quad (58)$$

forment une base de l'espace tangent  $T_p S_b^+$  en  $p$  à  $S_b^+$ . Dans 58,  $(\varphi_b^+)_x$  et  $(\varphi_b^+)_y$  désignent respectivement les dérivées partielles en  $x$  et  $y$  de  $\varphi_b^+$ . Pour obtenir la métrique riemannienne sur la surface  $S_b^+$ , il suffit de restreindre le produit scalaire euclidien<sup>2</sup> de  $\mathbb{R}^3$  à

2. Il est parfois utile de pondérer ce produit euclidien.

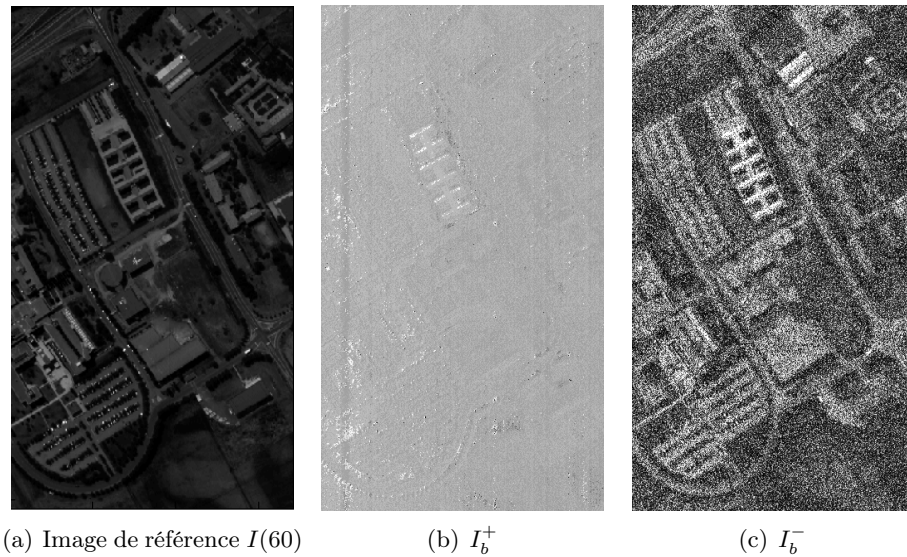


FIGURE 21 – Conversion décimale du codage binaire pour l'image PaviaU.

chacun des espaces tangents  $T_p S_b^+$ . La matrice de la forme bilinéaire symétrique sur  $T_p S_b^+$  qui donne la métrique cherchée s'écrit

$$G_b^+(x, y) = \begin{pmatrix} 1 + (I_b^+)_x^2 & (I_b^+)_x (I_b^+)_y \\ (I_b^+)_x (I_b^+)_y & 1 + (I_b^+)_y^2 \end{pmatrix} \quad (59)$$

les dérivées partielles des coefficients de  $G_b^+(x, y)$  étant évalués au point  $(x, y)$  de  $\Omega$ . La métrique  $G_b^+(x, y)$  permet de calculer toutes les informations métriques intrinsèques de la surface  $S_b^+$ . Soient  $(\nu_b^+)^+$  et  $(\nu_b^+)^-$  les deux valeurs propres réelles de  $G_b^+(x, y)$  avec  $(\nu_b^+)^+ \geq (\nu_b^+)^-$ . Pour détecter les contours dans l'image  $I_b^+$  il faut sélectionner les points de  $\Omega$  (ou les points de  $S_b^+$ ) pour lesquels la différence entre ces deux valeurs propres est grande. On introduit pour cela un indicateur<sup>3</sup> de force de contours

$$\varpi(x, y) = \sqrt{(\nu_b^+)^+ - (\nu_b^+)^-}. \quad (60)$$

La visualisation des contours se fait finalement via un seuillage par hystérésis de  $\varpi$ , [56].

### 3.1.4 Exemples de détection de contours

Nous donnons des exemples de résultats de détection de contours pour des images  $I_b^+$ . Comme mentionné ci-dessus, la détection de contours résulte d'un seuillage par hystérésis, c'est-à-dire à l'aide de deux seuils  $t_l$  et  $t_h$ . Les pixels au dessus du seuil haut  $t_h$  sont des points de contour significatifs, ils sont complétés par les pixels compris entre les deux seuils, pour autant qu'ils soient connexes aux premiers, considérés comme des points de contour moins significatifs. La connexité est définie par l'appartenance au voisinage. Nous avons pris

3. D'autres choix de combinaison entre les valeurs propres sont possibles.



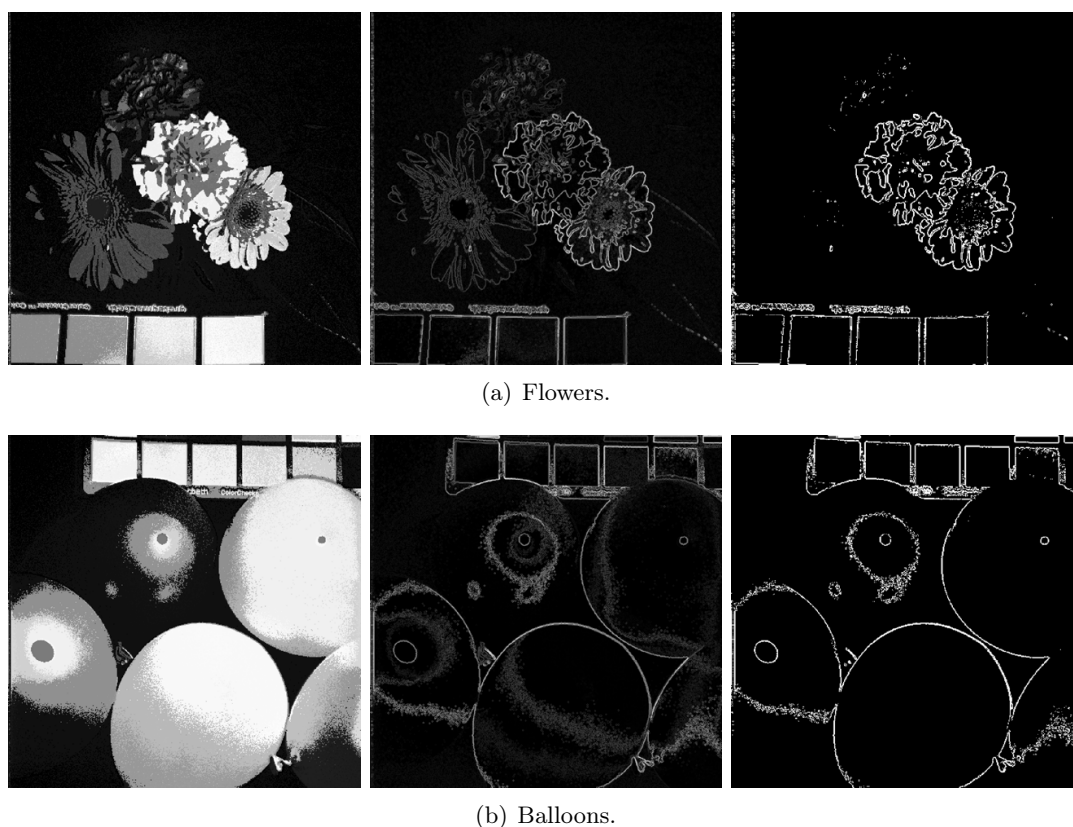


FIGURE 22 – Conversion décimale et détection de contours : image  $I_b^+$  (à gauche), indicateur de force  $\varpi$  (au centre) et contours détectés (à droite).

la valeur par défaut : 8 voisins. Pour le choix des seuils, nous avons pris les valeurs de  $I_b^+$  telles que au moins une proportion  $p_l$  et  $p_h$  leur soit supérieure. La figure 22 montre les résultats que l'on obtient pour les images "Flowers" et "Balloons", avec  $t_l = 0.9$  et  $t_h = 0.96$ , avec plus ou moins de succès puisque les paramètres et seuils n'ont pas été minutieusement ajustés.

Les résultats de la détection de contours montrent que le codage binaire introduit est effectivement un outil simple qui permet de rendre compte du contenu d'images hyperspectrales. Cependant, pour obtenir une segmentation vraiment significative, il est nécessaire de travailler selon l'approche décrite dans l'item 2 de la remarque 3, c'est-à-dire en utilisant les codes binaires en tant que tels.

### 3.2 Distance de Hamming généralisée

Nous reprenons ici les grandes lignes de l'article [9]. La problématique est de définir une distance entre chaînes de bits de même dimension. Nous notons  $\mathcal{B}(n)$  l'ensemble  $\{0, 1\}^n$ . La distance de Hamming usuelle entre deux éléments de  $\mathcal{B}(n)$  se calcule à partir du seul nombre de bits qui diffèrent dans les deux chaînes. Pour les applications qui nous concernent, il est essentiel de prendre en compte la proximité des différences. Ainsi les deux éléments

1100100000 et 1100010000 de  $\mathcal{B}(10)$  doivent être évalués plus proches entre eux que les deux éléments 1100100000 et 1100000001, bien que la distance de Hamming entre les deux premières chaînes et les deux dernières soit la même.

### 3.2.1 Définition de la distance

Utilisant pleinement le caractère binaire, seuls les 1-bits et leur place interviennent dans le calcul de la distance de Hamming généralisée. Ainsi à tout élément  $b$  de  $\mathcal{B}(n)$  on associe la suite <sup>4</sup>

$$\sigma(b) = (\sigma_1(b), \sigma_2(b), \dots, \sigma_{n(b)}(b)) \quad (61)$$

où  $\sigma_i(b)$  désigne la position du  $i$ -ème 1-bit de  $b$  pour  $1 \leq i \leq n(b)$ ,  $n(b)$  étant le nombre de 1-bits de  $b$ . Ainsi,  $\sigma(110) = (1, 2)$  et  $\sigma(011) = (2, 3)$ . Le calcul de la distance de Hamming généralisée entre deux éléments  $b_1$  et  $b_2$  de  $\mathcal{B}(n)$  se fait de façon dynamique en évaluant les coûts minimaux  $c(i, j)$  nécessaires pour transformer les  $i$  premiers 1-bits de  $b_1$  en les  $j$  premiers 1-bits de  $b_2$  pour  $1 \leq i \leq n(b_1)$  et  $1 \leq j \leq n(b_2)$ .

**Définition 8.** *La distance de Hamming généralisée entre deux chaînes  $b_1$  et  $b_2$  de  $\mathcal{B}(n)$  est par définition*

$$DHG(b_1, b_2) = c(n(b_1), n(b_2)) \quad (62)$$

*c'est-à-dire le coût minimal nécessaire pour transformer les  $n(b_1)$  1-bits de  $b_1$  en les  $n(b_2)$  1-bits de  $b_2$ .*

Les transformations utilisées pour calculer les coûts  $c(i, j)$  sont de trois types.

1. L'insertion d'un 1-bit. On souhaite passer de la suite  $(\sigma_1(b_1), \sigma_2(b_1), \dots, \sigma_i(b_1))$  à la suite  $(\sigma_1(b_2), \sigma_2(b_2), \dots, \sigma_j(b_2))$  en insérant un 1-bit. Pour ce faire, il est nécessaire que  $\sigma_i(b_1)$  soit strictement inférieur à  $\sigma_j(b_2)$ . Considérons à titre d'exemple les deux éléments de  $\mathcal{B}(7)$   $b_1 = 1010111$  et  $b_2 = 1011111$  avec  $\sigma_2(b_1) = 3$  et  $\sigma_3(b_2) = 4$  ( $i = 2$  et  $j = 3$ ). Le coût  $c(2, 3)$  pour passer des 2 premiers 1-bits de  $b_1$  aux trois premiers 1-bits de  $b_2$  en insérant un 1-bit est le coût pour passer de la chaîne 101 à la chaîne 1011.

En notant  $c_I$  le coût d'insertion d'un 1-bit, on a :

$$c(i, j) = c_I + c(i, j - 1) \quad (63)$$

2. La suppression d'un 1-bit. C'est la situation "symétrique" de la précédente. En inversant les rôles de  $b_1$  et  $b_2$ , le coût  $c(3, 2)$  est le coût nécessaire pour passer par suppression d'un 1-bit de la chaîne 1011 à la chaîne 101. Pour cette opération, on doit avoir  $\sigma_i(b_1)$  strictement supérieur à  $\sigma_j(b_2)$ .

En notant  $c_D$  le coût de suppression d'un 1-bit, on a :

$$c(i, j) = c_D + c(i - 1, j) \quad (64)$$

---

4. Cette suite est une façon très condensée de représenter l'information initiale contenue dans l'image hyperspectrale.

3. Le “shift”<sup>5</sup>. Considérons les éléments  $b_1 = 1010011$  et  $b_2 = 1011001$  de  $\mathcal{B}(7)$ . Pour transformer les 3 premiers 1-bits de  $b_1$  en les trois premiers 1-bits de  $b_2$ , on peut déplacer le troisième 1 bit de  $b_1$  de deux places vers la gauche. Dans ce cas  $\sigma_3(b_1) = 6$  et  $\sigma_3(b_2) = 4$ . La longueur du déplacement correspond à  $\sigma_3(b_1) - \sigma_3(b_2)$ .

En notant  $c_S (|\sigma_i(b_1) - \sigma_j(b_2)|)$  le coût du shift, on a :

$$c(i, j) = c_S (|\sigma_i(b_1) - \sigma_j(b_2)|) + c(i - 1, j - 1) \quad (65)$$

Les coûts  $c_I$  et  $c_D$  sont des constantes. Pour le coût  $c_S$  plusieurs choix sont possibles. Dans la suite nous travaillons avec un coût linéaire :

$$c_S (|\sigma_i(b_1) - \sigma_j(b_2)|) = \alpha_S |\sigma_i(b_1) - \sigma_j(b_2)| \quad (66)$$

où  $\alpha_S$  est une constante. Nous renvoyons à [9] pour des pistes d’ajustement de ce paramètre crucial en fonction du contexte.

### 3.2.2 Calcul de la distance

L’implémentation du calcul effectif de la distance de Hamming généralisée se fait de façon dynamique en calculant les coefficients de la matrice suivante

$$D(b_1, b_2) = \begin{pmatrix} d(0, 0) & d(0, 1) & \cdots & d(0, n(b_1)) \\ d(1, 0) & d(1, 1) & \cdots & d(1, n(b_1)) \\ \cdots & \cdots & \cdots & \cdots \\ d(n(b_2), 0) & d(n(b_2), 1) & \cdots & d(n(b_2), n(b_1)) \end{pmatrix} \quad (67)$$

où  $d(j, i) = c(i, j)$ . Pour ce faire, on “remplit” la matrice ligne par ligne en commençant chaque ligne par la gauche. Illustrons la démarche sur l’exemple des deux chaînes  $b_1 = 011$  et  $b_2 = 110$  avec des coûts  $c_I = 20$ ,  $c_D = 30$  et  $\alpha_S = 10$ .

Comme  $d(0, i) = c(i, 0) = ic_D$  et  $d(j, 0) = c(0, j) = jc_I$  on obtient tout d’abord

$$D(b_1, b_2) = \begin{pmatrix} 0 & 30 & 60 \\ 20 & & \\ 40 & & \end{pmatrix} \quad (68)$$

Calculons le coefficient  $d(1, 1) = c(1, 1)$ . Il s’agit de transformer la chaîne 01 correspondant à la suite (2) ( $\sigma_1(b_1) = 2$ ) en la chaîne 1 correspondant à la suite (1) ( $\sigma_1(b_2) = 1$ ).

On a donc  $\sigma_1(b_1) > \sigma_1(b_2)$ . Deux choix sont possibles.

1. On supprime le 1-bit de  $b_1$ , d’où :  $c(1, 1) = c_D + c(0, 1)$  soit encore  $d(1, 1) = 30 + d(1, 0) = 50$ .
2. On effectue un shift d’une place vers la gauche, d’où  $c(1, 1) = 10 + c(0, 0)$  soit encore  $d(1, 1) = 10 + d(0, 0) = 10$

Le coût minimal correspond évidemment au second choix.

5. C’est cette dernière transformation qui donne toute sa pertinence à la distance de Hamming dans sa version généralisée.

On obtient donc

$$D(b_1, b_2) = \begin{pmatrix} 0 & 30 & 60 \\ 20 & 10 & \\ 40 & & \end{pmatrix} \quad (69)$$

Calculons maintenant le coefficient  $d(1, 2) = c(2, 1)$ . Il s'agit de transformer la chaîne 011 correspondant à la suite (2,3) ( $\sigma_2(b_1) = 3$ ) en la chaîne 1 correspondant à la suite (1) ( $\sigma_1(b_2) = 1$ ).

- On a donc  $\sigma_2(b_1) > \sigma_1(b_2)$ . Deux choix sont possibles.
1. On supprime le 1-bit de  $b_1$ , d'où :  $c(2, 1) = c_D + c(1, 1)$  soit encore  $d(1, 2) = 30 + d(1, 1) = 40$ .
  2. On effectue un shift de deux places vers la gauche, d'où  $c(2, 1) = 10x2 + c(1, 0)$  soit encore  $d(1, 2) = 20 + d(0, 1) = 50$
- Le coût minimal correspond cette fois-ci au premier choix.

On obtient donc

$$D(b_1, b_2) = \begin{pmatrix} 0 & 30 & 60 \\ 20 & 10 & 40 \\ 40 & & \end{pmatrix} \quad (70)$$

Calculons le coefficient  $d(2, 1) = c(1, 2)$ . Il s'agit de transformer la chaîne 01 correspondant à la suite (2) ( $\sigma_1(b_1) = 2$ ) en la chaîne 11 correspondant à la suite (1,2) ( $\sigma_2(b_2) = 2$ ).

- On a donc  $\sigma_1(b_1) = \sigma_2(b_2)$ . Comme le premier 1-bit de  $b_1$  est aligné avec le second 1-bit de  $b_2$ , il n'y a rien à faire et  $c(1, 2) = d(2, 1) = 0 + c(0, 1) = d(1, 0) = 20$ .

On obtient donc

$$D(b_1, b_2) = \begin{pmatrix} 0 & 30 & 60 \\ 20 & 10 & 40 \\ 40 & 20 & \end{pmatrix} \quad (71)$$

Calculons enfin  $d(2, 2) = c(2, 2) = DHG(b_1, b_2)$ . Il s'agit de transformer  $b_1$  correspondant à la suite (2,3) ( $\sigma_2(b_1) = 3$ ) en la chaîne  $b_2$  correspondant à la suite (1,2) ( $\sigma_2(b_2) = 2$ ).

- On a donc  $\sigma_2(b_1) > \sigma_2(b_2)$ . Deux choix sont possibles.
1. On supprime le 1-bit de  $b_1$ , d'où :  $c(2, 2) = c_D + c(1, 2)$  soit encore  $d(2, 2) = 30 + d(2, 1) = 50$ .
  2. On effectue un shift d'une place vers la gauche, d'où  $c(2, 2) = 10 + c(1, 1)$  soit encore  $d(2, 2) = 10 + d(1, 1) = 20$
- Le coût minimal correspond au second choix.

On obtient donc

$$D(b_1, b_2) = \begin{pmatrix} 0 & 30 & 60 \\ 20 & 10 & 40 \\ 40 & 20 & 20 \end{pmatrix} \quad (72)$$

Finalement,  $DHG(b_1, b_2) = 20$ .



De façon générale, le remplissage de la matrice  $D(b_1, b_2)$  nécessite de distinguer trois cas.

1. Si  $\sigma_i(b_1) = \sigma_j(b_2)$  alors

$$c(i, j) = d(j, i) = c(i - 1, j - 1) \quad (73)$$

2. Si  $\sigma_i(b_1) > \sigma_j(b_2)$  alors

$$c(i, j) = d(j, i) = \min [c_D + c(i - 1, j), c_S (\sigma_i(b_1) - \sigma_j(b_2)) + c(i - 1, j - 1)] \quad (74)$$

3. Si  $\sigma_i(b_1) < \sigma_j(b_2)$  alors

$$c(i, j) = d(j, i) = \min [c_I + c(i, j - 1), c_S (\sigma_j(b_2) - \sigma_i(b_1)) + c(i - 1, j - 1)] \quad (75)$$

### 3.2.3 Un exemple simple

Le coût du shift est linéaire de coefficient  $\alpha_S$ . Nous traitons ici un exemple simple qui montre l'influence du paramètre  $\alpha_S$  dans le calcul de la distance de Hamming généralisée.

**Exemple 2.** *Considérons les trois chaînes  $a_1 = 1100$ ,  $a_2 = 1010$  et  $a_3 = 1001$  de  $\mathcal{B}(5)$ . Les paramètres choisis pour le calcul de la distance sont  $c_i = c_d = 1$  et  $\alpha_S = 1,6$ . Si l'on ne tient pas compte du shift, la distance entre les chaînes  $a_1$  et  $a_2$  et  $a_1$  et  $a_3$  est donnée par les opérations suivantes :*

$$\begin{array}{l} 1100 \xrightarrow{d_2} 1000 \xrightarrow{i_3} 1010 \\ 1100 \xrightarrow{d_2} 1000 \xrightarrow{i_4} 1001 \end{array} \quad (76)$$

*et le coût minimal est de 2 dans les deux cas. Le résultat n'est autre que la distance de Hamming usuelle. En tenant compte du shift et en utilisant les opérations suivantes :*

$$\begin{array}{l} 1100 \xrightarrow{2 \leftrightarrow 3} 1010 \\ 1100 \xrightarrow{2 \leftrightarrow 3} 1010 \xrightarrow{3 \leftrightarrow 4} 1001 \end{array} \quad (77)$$

*le coût est de 1,6 dans le premier cas et de 3,2 dans le second. Ceci signifie que  $DHG(a_1, a_2) = 1,6$  et  $DHG(a_1, a_3) = 2$  : conformément à ce que l'on veut mettre en évidence  $a_2$  est plus proche de  $a_1$  que  $a_3$  ne l'est.*

Dans cet exemple, si  $\alpha_S$  est plus grand que 2, alors la distance de Hamming généralisée coïncide avec la distance de Hamming usuelle.

## 3.3 Segmentation par clustering

Dans cette section nous exploitons le codage binaire de la définition 7 pour segmenter des images hyperspectrales à partir des formes des profils spectraux. La distance de Hamming généralisée (62) s'interprète dans ce contexte comme une distance entre ces dernières. La segmentation est obtenue à partir d'un algorithme de  $k$ -modes comparable à celui de [31]. Il s'agit essentiellement d'une adaptation de l'algorithme des  $k$ -means au cas de données binaires, et nous renvoyons le lecteur à la section 1.4 de l'introduction pour des rappels sur la classification non supervisée par clustering et en particulier par  $k$ -means (page 18).

### 3.3.1 Algorithmes $k$ -modes

Le problème de la classification de données binaires n'est pas sans poser quelques difficultés, voir par exemple [58]. En particulier les algorithmes classiques de type  $k$ -means nécessitant des descentes de gradients pour la recherche des centres ne respectent pas le caractère catégoriel, et *a fortiori* binaire des données<sup>6</sup>. Nous proposons ici une méthode simple et originale de recherche de  $k$ -modes stricts. Rappelons que nous considérons une image  $I$  de taille spatiale  $M \times N$  et de taille spectrale  $\Lambda$ , et que pour chaque pixel  $(m, n)$ , le codage binaire fournit un vecteur  $I_b(m, n)$  de dimension  $p = \Lambda - 1$  qui est un objet à classifier de l'ensemble

$$X = \{I_b(m, n) \in \{0, 1\}^p, (m, n) \in E\} = \{x_i \in \{0, 1\}^p, 1 \leq i \leq M \times N\} \quad (78)$$

de cardinal  $M \times N$ . Les deux principales modifications à apporter à l'algorithme des  $k$ -means, page 18, sont les suivantes.

1. Utiliser une distance de correspondances pour données catégorielles ; en l'occurrence, nous utilisons la distance Hamming généralisée  $DHG$  définie par (62).
2. Remplacer les moyennes par des modes fréquentiels, dont chaque composante est (tout comme pour les objets) à valeur dans l'ensemble des catégories possibles de chaque attribut  $dom_l = \{a_1^l, \dots, a_{n_l}^l\}$ ,  $1 \leq l \leq p$ ; pour le cas particulier des données binaires, ces domaines sont tous les mêmes, à savoir  $dom_l = \{0, 1\}$ ,  $1 \leq l \leq p$ .

La fonctionnelle (11) à minimiser devient

$$J_{DHG}(U, C) = \sum_{i=1}^n \sum_{j=1}^k u_{ik} DHG(x_i, c_j) \quad (79)$$

où  $u_{ij} \in \{0, 1\}$  représente l'appartenance de l'objet  $x_i$  au cluster  $j$  représenté par son mode  $c_j$ , appartenances obtenues sous les mêmes contraintes (12-13) que pour les  $k$ -means. Notons que les modes ne sont alors pas obligatoirement des points de  $X$ , mais leur actualisation par l'équation (81) ci-après assure que la fonctionnelle (79) est minimale à l'itération  $t$ , étant donnée la matrice de partition courante  $U^{(t)}$  (voir preuve dans [32]). À partir de modes initiaux choisis aléatoirement dans  $X$ , l'algorithme alterne les phases d'actualisation de la matrice de partition et de l'ensemble des modes.

0. Initialiser  $C^{(0)}$  (ou  $U^{(0)}$  et inverser les deux étapes d'actualisation), et le paramètre d'arrêt  $r^{(0)}$  à une valeur  $\varepsilon$  petite.
1. Calculer les distances entre les objets  $X$  et les centres  $C^{(t)}$  selon (62).
2. Actualiser  $U^{(t)} : \forall 1 \leq i \leq n, \forall 1 \leq j \leq k$ ,

$$u_{ij}^{(t)} = \begin{cases} 1 & \text{si } j = \underset{j=1,k}{\operatorname{argmin}} DHG(x_i, c_j^{(t)}) \\ 0 & \text{sinon} \end{cases} \quad (80)$$

---

6. Le problème, par exemple, de la définition d'un barycentre de Karcher dans notre contexte semble largement ouvert.

3. Actualiser  $C^{(t)} : \forall 1 \leq j \leq k, \forall 1 \leq l \leq p,$

$$c_{jl}^{(t)} = \operatorname{argmax}_{x_i: u_{ij}^{(t)}=1} x_{il} \quad (81)$$

dans le cas binaire

$$c_{jl}^{(t)} = \begin{cases} 1 & \text{si } \sum_{i=1}^n u_{ij} x_{il} > \sum_{i=1}^n u_{ij} (1 - x_{il}) \\ 0 & \text{sinon} \end{cases} \quad (82)$$

4. Tester la stabilité de la partition <sup>7</sup> :

$$r^{(t)} = \|U^{(t)} - U^{(t-1)}\| \quad (83)$$

aller en 1 tant que  $r^{(t)} \geq \varepsilon$ .

**Remarque 4.** *L'étape d'initialisation influe fortement sur la convergence de l'algorithme<sup>8</sup> aux niveaux :*

- *De l'optimalité du couple  $(U, C)$  final qui ne réalise qu'un minimum local de la distance intra-clusters  $J_{DHG}$  (79). Il est par conséquent d'usage d'exécuter un certain nombre de fois l'algorithme pour ne retenir que la meilleure solution.*
- *Du temps d'exécution.*

Des travaux portent sur l'amélioration de cette étape consistant à proposer des méthodes efficaces d'initialisation aléatoire des modes selon diverses approches [37]. Nous décrivons ci-dessous une méthode permettant de choisir efficacement les  $k$  modes initiaux  $C^{(0)}$ .

Soit  $l$  un entier avec  $l \geq k$ . Soit  $x_1 = I_b(m_1, n_1)$ ,  $x_2 = I_b(m_2, n_2)$ , ...,  $x_l = I_b(m_l, n_l)$ ,  $l$  codes binaires de pixels tirés aléatoirement et  $X = \{x_i, 1 \leq i \leq l\}$ .

**Définition 9.** *Un sous-ensemble  $A = \{x_{i_j}, 1 \leq j \leq k\}$  de l'ensemble  $X$  des pixels tirés aléatoirement est dit  $k$ -maximal pour la distance de Hamming généralisée si : pour tout  $x$  de  $X$  n'appartenant pas à  $A$ , il existe un  $x_{i_0}$  de  $A$  tel que*

$$DHG(x, x_{i_0}) \leq \min_{1 \leq j_1 \neq j_2 \leq k} DHG(x_{i_{j_1}}, x_{i_{j_2}}) \quad (84)$$

Voici comment obtenir pratiquement un ensemble  $A$   $k$ -maximal pour la distance de Hamming généralisée à partir de l'ensemble  $X$ .

**Proposition 1.** *Soit  $X = \{x_1, x_2, \dots, x_l\}$  un ensemble de codes binaires. Le sous-ensemble*

$$A = \left\{ x_i \in X, \exists j \in \{1, 2, \dots, l\}, L_{l-k}(i, j) \times L_{l-k}(j, i) < \mu^2 \right\} \quad (85)$$

*où  $\mu$  et les matrices  $L_p$  de taille  $l \times l$  sont définis comme expliqué ci-dessous est  $k$ -maximal pour la distance de Hamming généralisée (62).*

7. Dans (83),  $\|\cdot\|$  désigne n'importe quelle norme matricielle, par exemple  $\|A\| = \max_i \max_j a_{ij}$ .

8. Ceci vaut aussi pour les  $k$ -means, voir [33].

La matrice  $L_0$  est donnée par

$$L_0(i, j) = \begin{cases} GHD(x_i, x_j) & \text{si } i < j \\ \mu := 1 + \max_{1 \leq i, j \leq l} GHD(x_i, x_j) & \text{sinon} \end{cases} \quad (86)$$

Notons

$$(\alpha_0, \beta_0) = \underset{1 \leq i, j \leq l}{\operatorname{argmin}} L_0(i, j) \quad (87)$$

Les matrices  $L_p$  sont définies récursivement de la façon suivante. La matrice  $L_{p-1}$  étant donnée, soit  $s_{p-1}$  au choix égal à  $\alpha_{p-1}$  ou  $\beta_{p-1}$  avec

$$(\alpha_{p-1}, \beta_{p-1}) = \underset{1 \leq i, j \leq l}{\operatorname{argmin}} \{L_{p-1}(i, j)\} \quad (88)$$

Alors,  $L_p$  est définie par

$$L_p(i, j) = \begin{cases} \mu & \text{si } i = s_{p-1} \text{ ou } j = s_{p-1} \\ L_{p-1}(i, j) & \text{sinon} \end{cases} \quad (89)$$

### 3.3.2 Illustrations

La figure 23 illustre les résultats que l'obtient sur les images "Flowers" et "Balloons" lorsque  $k = 5$  et  $\alpha_S = 2$  et  $\alpha_S = 0, 4$ . La figure 24 illustre sur l'image "PaviaU" l'évolution de la segmentation lorsqu'on augmente le nombre  $k$  de clusters. L'évolution de la segmentation en fonction du paramètre du shift  $\alpha_S$ , voir la figure 25, montre que plus le  $\alpha_S$  devient grand, et donc plus la distance de Hamming généralisée se rapproche de la distance de Hamming usuelle, plus l'image obtenue est bruitée. L'expérience montre qu'il convient généralement de choisir  $\alpha_S$  plus petit que 1.

**Remarque 5.** *L'aspect visuellement bruité des résultats est tout à fait logique. En effet, aucune information spatiale n'étant utilisée, il y a un nombre important de petites zones de pixels ayant des variations de la forme de leur profil spectral similaires à celles de zones plus conséquentes mais distantes.*

### 3.3.3 Amélioration du codage binaire

Il existe bien des possibilités pour améliorer le codage binaire de la définition 7. Nous allons traiter ici un exemple qui permet d'obtenir des résultats sensiblement meilleurs. Pour cela, introduisons quelques nouvelles notations.

Étant donné  $A$  un ensemble de  $\Lambda - 1$  éléments et  $\delta$  un entier inférieur ou égal à  $\Lambda - 1$ , nous notons

$$\min_\delta(A) = \{a_1, \dots, a_\delta\} \quad (90)$$

avec  $a_1 \leq a_2 \leq \dots \leq a_\delta \leq a$  pour tout élément  $a$  de  $A \setminus \{a_1, \dots, a_\delta\}$ . De même,

$$\max_\delta(A) = \{a_1, \dots, a_\delta\} \quad (91)$$

avec  $a_1 \geq a_2 \geq \dots \geq a_\delta \geq a$  pour tout élément  $a$  de  $A \setminus \{a_1, \dots, a_\delta\}$ . Soit

$$\Delta I = \{\Delta I(m, n)(\lambda + 1), 1 \leq \lambda \leq \Lambda - 1\} \quad (92)$$

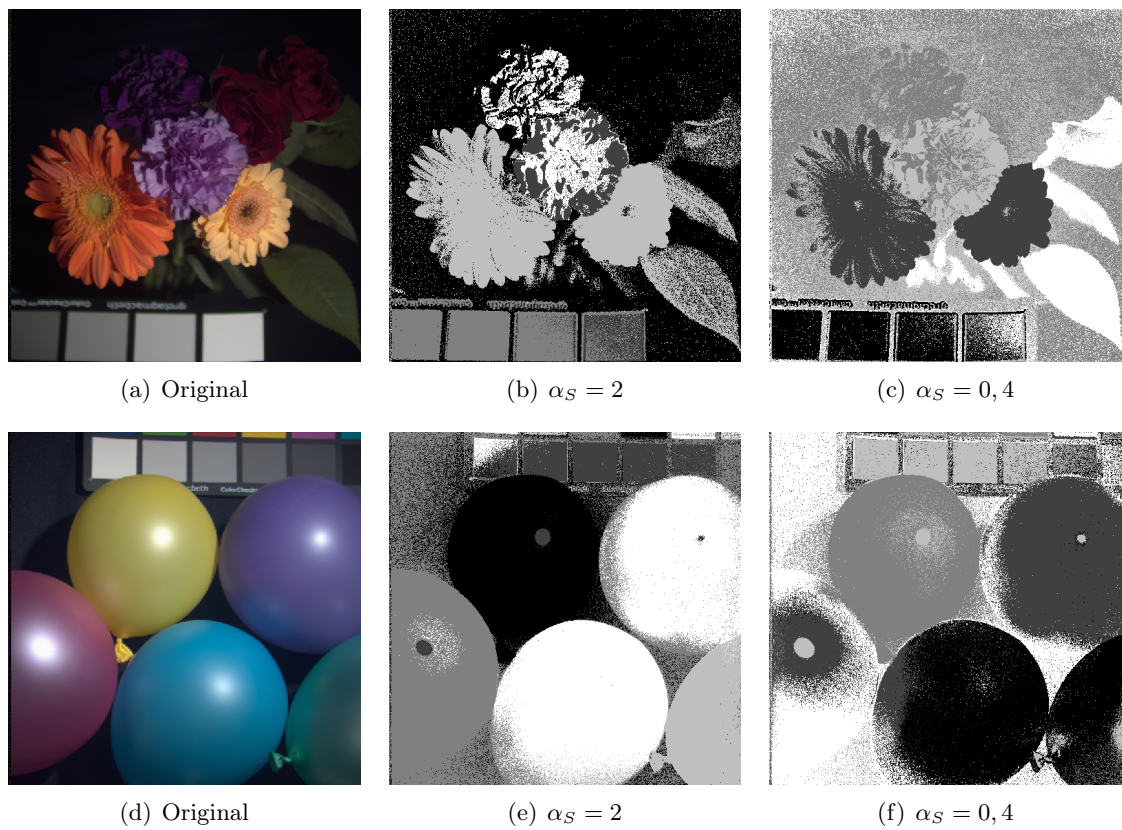


FIGURE 23 – Exemples de résultats de l’algorithme de  $k$ -modes avec  $k = 5$  et deux valeurs différentes pour le coût du shift  $\alpha_S$  : Flowers (première ligne) et Balloons (seconde ligne).



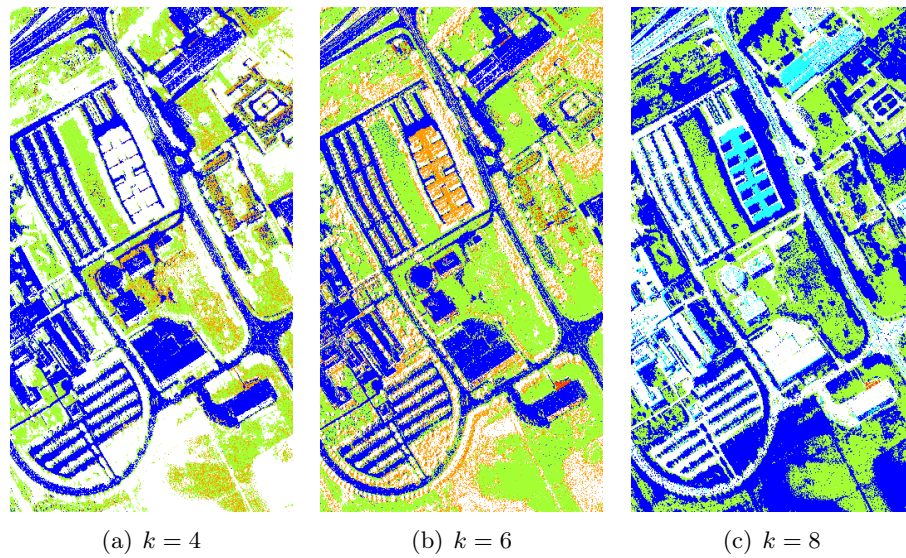


FIGURE 24 – Exemples de segmentation par  $k$ -modes sur l'image PaviaU pour différentes valeurs de  $k$  et le paramètre du shift  $\alpha_S$  fixé à 0,2.

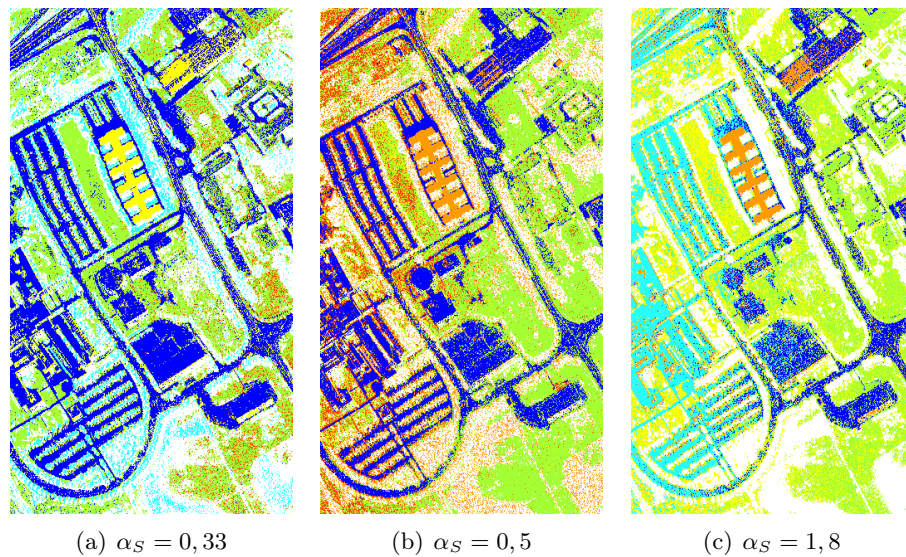


FIGURE 25 – Exemples de segmentation par  $k$ -modes sur l'image PaviaU pour différentes valeurs du paramètre du shift  $\alpha_S$  et  $k$  fixé à 6.

**Définition 10.** Les codages binaires  $I_{b,+,\delta}$  et  $I_{b,-,\delta}$  de l'image hyperspectrale  $I$  sont les deux applications de  $E$  à valeurs dans  $\{0, 1\}^{\Lambda-1}$  définies par :

$$I_{b,+,\delta}(m, n)(\lambda) = \begin{cases} 1 & \text{si } \Delta I(m, n)(\lambda + 1) \in \max_{\delta}(\Delta I) \\ 0 & \text{sinon} \end{cases} \quad (93)$$

$$I_{b,-,\delta}(m, n)(\lambda) = \begin{cases} 1 & \text{si } \Delta I(m, n)(\lambda + 1) \in \min_{\delta}(\Delta I) \\ 0 & \text{sinon} \end{cases} \quad (94)$$

avec  $\delta < \Lambda - 1$  et  $1 \leq \lambda \leq \Lambda - 1$ .

Pour la mise en pratique, les seules modifications à apporter à l'algorithme des  $k$ -modes binaires décrit page 53 sont les suivantes.

- Considérer deux ensembles de  $N \times M$  objets  $X_{+,\delta}$  et  $X_{-,\delta}$  (au lieu de  $X$ ) composés des codes binaires  $I_{b,+,\delta}(m, n)(\lambda)$  et  $I_{b,-,\delta}(m, n)(\lambda)$  (au lieu de  $I_b(m, n)$ ) de dimension  $p = \Lambda - 1$
- Initialiser, puis rechercher deux ensembles de modes  $C_{+,\delta}$  et  $C_{-,\delta}$  correspondants.
- Utiliser comme distance entre les objets  $x_i = (x_{i,+,\delta}, x_{i,-,\delta})$  à classifier et modes binaires  $c_j = (c_{j,+,\delta}, c_{j,-,\delta})$  une combinaison des distances de Hamming généralisées :

$$DHG(x_i, c_j) = GHD(x_{i,+,\delta}, c_{j,+,\delta}) + GHD(x_{i,-,\delta}, c_{j,-,\delta}) \quad (95)$$

Le reste de l'algorithme demeure inchangé. Les résultats obtenus avec ce nouvel algorithme sont sensiblement meilleurs que les précédents. En particulier, le nouveau codage binaire permet de détecter une tâche présente dans la bande échantillon de "PaviaU" qui était moins bien détectée auparavant (voir la figure 26).

### 3.3.4 Comparaison avec des méthodes de référence

Comparer nos résultats avec des résultats de segmentation de la littérature n'est pas simple car si elle abonde dans le domaine de la classification supervisée bien peu d'articles :

- envisagent la segmentation par classification totalement non supervisée des pixels ; les approches avec supervision partielle ou combinant une méthode non supervisée et une méthode supervisée contraignent bien évidemment le processus de manière favorable ;
- proposent une méthode n'utilisant que l'information spectrale ; souvent, des contraintes spatiales sont ajoutées ;
- utilisent l'ensemble complet des bandes spectrales, par exemple  $\Lambda = 103$  pour notre image de référence l'image "Pavia University Scene" ; souvent soit un échantillonnage est effectué, soit la dimensionalité est réduite en amont par exemple par composantes principales ;
- montrent enfin des résultats pour le même nombre de classes ; très rarement même celui de la vérité-terrain dont nous avons déjà montré les limites.

Nous avons choisi de nous en tenir aux méthodes de classification totalement non supervisée de pixels sans information spatiale, et reportons à la figure 27 des résultats trouvés dans [13, 72] obtenus en utilisant les méthodes suivantes.

1. ISODATA, [2], est un algorithme de type moindres carrés qui permet de séparer les classes si elles sont trop dispersées, de les fusionner si elles sont trop proches ou

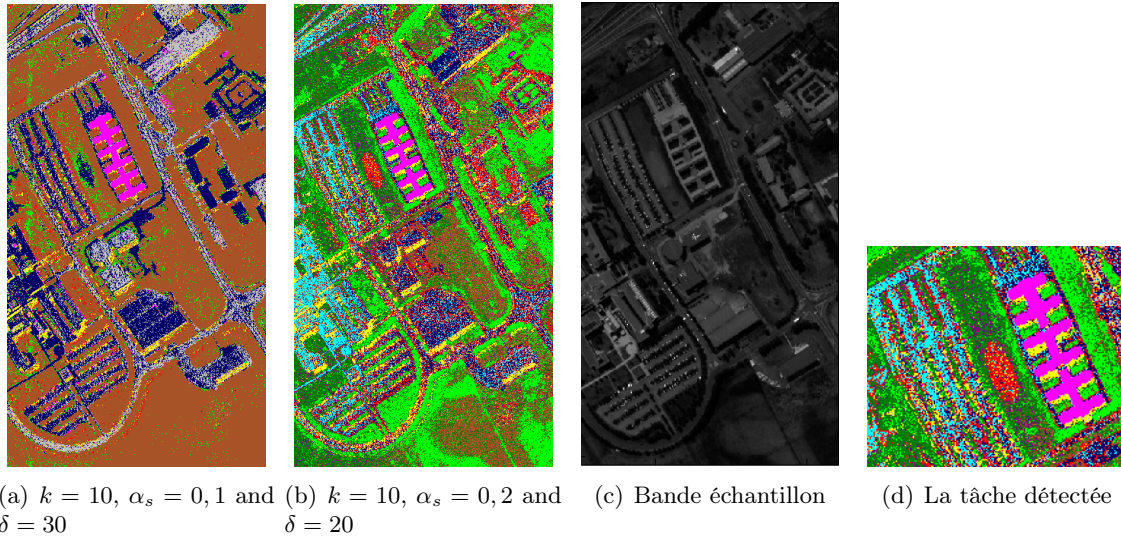


FIGURE 26 – Résultats obtenus à partir de l’algorithme de  $k$ -modes modifié avec  $k = 10$  et deux valeurs différentes du paramètre  $\delta$  : (a) and (b). La bande échantillon de “PaviaU” et la tâche détectée : (c) and (d).

encore de ne pas les prendre en compte si elles contiennent trop peu d’objets. Seul le meilleur résultat de 10 essais a été retenu pour obtenir finalement 9 classes, 27(a).

2. EM (Expectation Maximization, [18]) est un algorithme qui présuppose que les données correspondent à un mélange de  $k$  distributions (gaussiennes) dont il s’agit d’identifier les paramètres. Compte tenu de son coût en temps de calcul, la segmentation en 10 classes 27(b) n’a pu être obtenue qu’en réduisant l’image initiale à une image de 10 bandes obtenues en moyennant les 10 bandes voisines.
3. AP (Affinity Propagation, [25]) est un algorithme qui affecte itérativement les pixels de l’image suivant qu’ils sont de bons représentants pour les autres pixels et sont choisis comme tels par les autres pixels. Ce type d’algorithme n’est pas adapté à la complexité des images hyperspectrales. Pour obtenir le résultat 27(c), 12 962 pixels ont été sélectionnés au hasard en entrée pour obtenir les 29 classes. Les 194 438 pixels restant ont été ensuite affectés à ces classes.
4. NPSEM, [10], est une version non paramétrique de l’algorithme EM stochastique, [15]. Le résultat de la segmentation 27(d) fait apparaître 17 classes.

Ces résultats montrent que la méthode proposée produit des segmentations relativement satisfaisantes, si on garde en mémoire qu’aucune information spatiale n’est utilisée. L’application d’un simple filtre médian en post-traitement permet d’améliorer très sensiblement la qualité de la segmentation, voir la figure 28.

### 3.4 Conclusions et perspectives du chapitre

Nous avons présenté dans ce chapitre une approche très simple à mettre en œuvre qui permet d’obtenir des résultats satisfaisants de segmentation non supervisée d’image hyperspectrales.



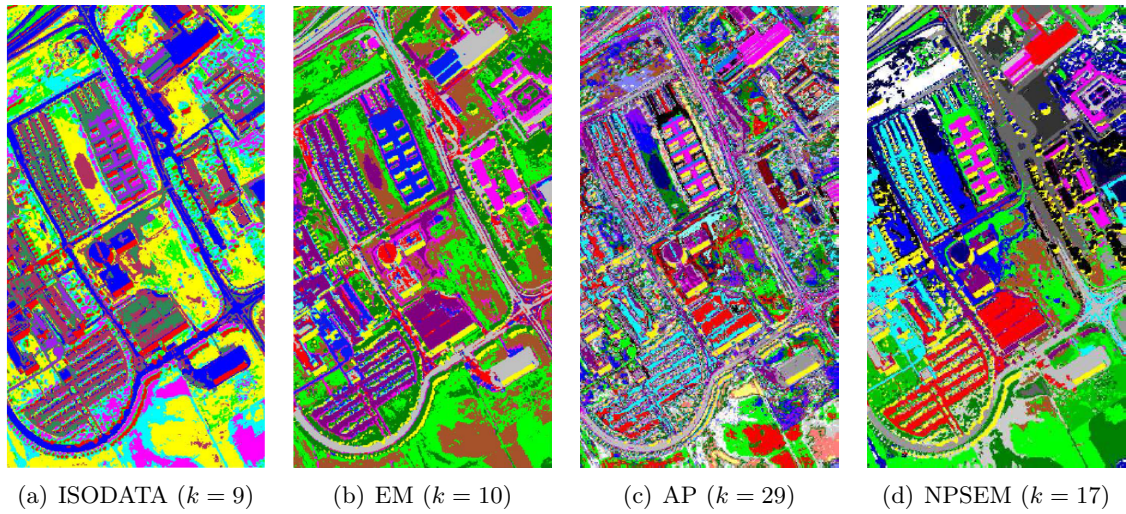
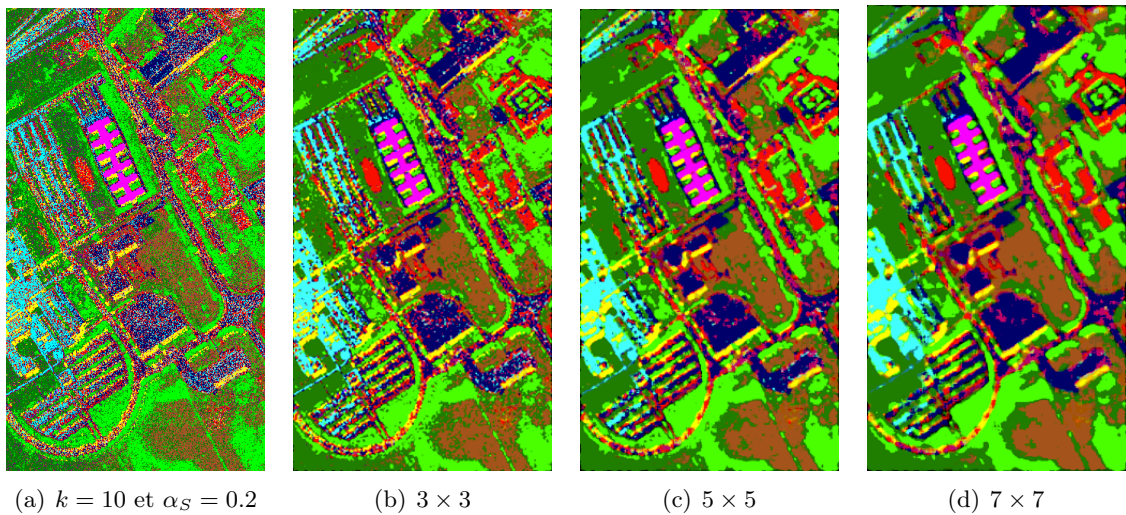


FIGURE 27 – Segmentation de l'image PaviaU par des méthodes de la littérature.

FIGURE 28 – Segmentation de l'image PaviaU avec  $k = 10$  et  $\alpha_S = 0.2$  et application d'un filtre médian de différentes tailles.

Il est important d'insister sur le fait que les segmentations proposées reposent essentiellement sur la donnée des suites  $\sigma(b)$  telles que définies par la formule (61) à partir du codage binaire ou de ses variantes éventuelles. Il s'agit d'une réduction très significative de la taille de l'information initiale : le profil spectral associé à chaque pixel est remplacé par une suite d'entiers qui dans la pratique est très courte. Par ailleurs, l'algorithme de  $k$ -modes proposé est peu coûteux ne nécessitant qu'un petit nombre de calculs de la distance de Hamming généralisée.

D'autres stratégies d'utilisation du codage binaire peuvent être envisagées. La plus naturelle est de reprendre l'approche de la segmentation par similarité proposée dans le chapitre précédent. La mesure de similarité spectrale est alors donnée par la distance de Hamming généralisée. Pour mettre en application cette idée, il est nécessaire de calculer les distances de Hamming généralisées entre tous les pixels de l'image hyperspectrale. Il en va de même si l'on choisit d'utiliser un clustering relationnel (voir [44]).

Il est également possible d'introduire des outils de la logique floue en étendant :

- directement l'algorithme des  $k$ -modes [32], ou
- la définition des codages binaires  $I_{b,+, \delta}(m, n)(\lambda)$  et  $I_{b,-, \delta}(m, n)(\lambda)$  au cas continu dans  $[0,1]$ , par exemple à partir des résultats de [50] sur la définition du  $k$ ème plus grand (ou plus petit) élément.

Signalons pour terminer que nous n'avons pas abordé la question du débruitage d'une chaîne binaire obtenue à partir de variations spectrales. A vrai dire, aucune solution évidente ne nous est apparue bien que plusieurs pistes puissent être explorées suivant des approches statistiques (géométrie de l'information) ou en utilisant des codes correcteurs par exemple.



## Chapitre 4

# Segmentation riemannienne

---

Comme nous l'avons déjà signalé dans l'introduction, il existe peu de travaux dédiés à l'analyse et au traitement d'images hyperspectrales exploitant la géométrie sous-jacente des images, c'est-à-dire les propriétés métriques des surfaces associées à ces dernières. C'est pourquoi nous explorons dans ce chapitre les possibilités d'utilisation d'outils de la géométrie riemannienne à des fins de segmentation. Comme c'est l'habitude dans ce type d'approches, la modélisation théorique se fait en considérant que les données images sont des applications lisses, c'est-à-dire suffisamment différentiables pour que les calculs effectués aient un sens. Ce n'est qu'au moment de l'implémentation effective que le caractère discret des données est pris en compte.

Une image hyperspectrale peut être vue soit comme une surface dans l'espace  $\mathbb{R}^{2+\Lambda}$  soit comme une famille de  $\Lambda$  surfaces de  $\mathbb{R}^3$  ou soit encore comme une variété de dimension 3 de l'espace  $\mathbb{R}^4$ . L'outil principal pour décrypter les propriétés métriques d'une variété plongée dans un espace euclidien est la première forme fondamentale, ou encore la métrique riemannienne. Il s'agit d'un champ de tenseurs que l'on peut considérer comme la donnée en tout point de la surface d'une matrice  $2 \times 2$  ou  $3 \times 3$  symétrique définie positive. C'est à partir de ces matrices que l'on calcule par exemple la distance géodésique ou la courbure de Gauss.

L'espace  $SPD(d)$  des matrices  $d \times d$  symétriques définies positives est lui-même une variété munie d'une métrique riemannienne explicite connue sous le nom de métrique de Rao ou métrique de Fisher. Cette métrique est pertinente dans bien des contextes, notamment en géométrie de l'information, car d'une part elle est associée à un groupe d'isométries (le groupe symplectique) qui agit naturellement sur l'espace de Siegel  $\mathbb{S}(d)$  et d'autre part elle correspond à la métrique de Fisher de l'espace des gaussiennes multivariées centrées. On dispose de plus d'une formule close pour la distance géodésique, ce qui permet de mesurer efficacement la distance entre deux matrices symétriques définies positives. Nombre de travaux plus ou moins récents sont consacrés à l'implémentation d'algorithmes de calcul de cette distance.

L'hypothèse de départ de ce qui suit est que la métrique riemannienne, ou la famille de  $\Lambda$  métriques riemanniennes, évaluée en point est un bon descripteur de l'image hyperspectrale et que de plus la distance de Rao-Fisher est pertinente sur l'espace de ces descripteurs. On se propose donc naturellement de tester un algorithme de classification, l'algorithme des  $k$ -means, ayant ces données en entrée. Ceci présuppose de disposer d'une bonne notion de

moyenne dans  $SPD(d)$ . Le choix le plus naturel est la moyenne de Karcher obtenue par minimisation et qui se prête bien à une estimation par descente de gradient, le gradient étant calculé à l'aide de la métrique de Rao de  $SPD(d)$ . Nous étudions également une seconde possibilité suivant [30] en adoptant un processus itératif exploitant le fait que l'on dispose de formules simples pour la paramétrisation des géodésiques. Les résultats obtenus montrent que les algorithmes de segmentation ainsi élaborés sont sensibles au bruit et qu'il est préférable de débruiter les données avant de les mettre en oeuvre. Nous proposons pour cela d'utiliser un filtre médian défini à partir de médianes géométriques dans  $SPD(d)$ .

## 4.1 Images hyperspectrales et variétés

Soit  $I$  une image hyperspectrale de taille spatiale  $M \times N$  et de taille spectrale  $\Lambda$ . Soit  $E = \{1, 2, \dots, M\} \times \{1, 2, \dots, N\}$  et  $W = \{1, 2, \dots, \Lambda\}$ .

À l'image  $I$  on associe une surface  $S$  de l'espace  $\mathbb{R}^{2+\Lambda}$  paramétrée par

$$\begin{aligned} \varphi : \Omega &\longrightarrow \mathbb{R}^{2+\Lambda} \\ \varphi(x, y) &= (x, y, I(x, y)) = (I^1(x, y), I^2(x, y), \dots, I^\Lambda(x, y)) \end{aligned} \quad (96)$$

où  $\Omega$  est le domaine spatial de l'image. Autrement dit  $S$  est le graphe dans  $\mathbb{R}^{2+\Lambda}$  de l'application  $I$ . Soit  $p = \varphi(x, y)$  un point de  $S$ . Comme nous l'avons déjà vu dans la section 3.1.3, les deux vecteurs

$$\varphi_x = \begin{pmatrix} 1 \\ 0 \\ I_x^1 \\ \vdots \\ I_x^\Lambda \end{pmatrix} \quad \varphi_y = \begin{pmatrix} 0 \\ 1 \\ I_y^1 \\ \vdots \\ I_y^\Lambda \end{pmatrix} \quad (97)$$

forment une base de l'espace tangent  $T_p S$  en  $p$  à  $S$ . Dans (97),  $\varphi_x$  et  $\varphi_y$  désignent respectivement les dérivées partielles en  $x$  et  $y$  de l'application  $\varphi$ . Pour obtenir la métrique riemannienne sur la surface  $S$ , il suffit de restreindre le produit scalaire euclidien de  $\mathbb{R}^{2+\Lambda}$  à chacun des espaces tangents  $T_p S$ ,  $p \in S$ . La matrice de la forme bilinéaire symétrique sur  $T_p S$  qui donne la métrique cherchée s'écrit

$$G(x, y) = \begin{pmatrix} 1 + \sum_{\lambda=1}^{\Lambda} (I_x^\lambda)^2 & \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_y^\lambda \\ \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_y^\lambda & 1 + \sum_{\lambda=1}^{\Lambda} (I_y^\lambda)^2 \end{pmatrix} \quad (98)$$

les dérivées partielles des coefficients de  $G(x, y)$  étant évalués au point  $(x, y)$  de  $\Omega$ . Par construction, la matrice  $G(x, y)$  est symétrique définie positive pour tout couple  $(x, y)$  de  $\Omega$ , i.e.  $G(x, y)$  appartient à  $SPD(2)$  pour tout  $(x, y)$  de  $\Omega$ .

On peut également considérer l'image hyperspectrale  $I$  comme la donnée de  $\Lambda$  surfaces  $S_\lambda$  de  $\mathbb{R}^3$  paramétrées par

$$\begin{aligned} \varphi_\lambda : \Omega &\longrightarrow \mathbb{R}^3 \\ \varphi_\lambda(x, y) &= (x, y, I^\lambda(x, y)) \end{aligned} \quad (99)$$

et considérer les  $\Lambda$  métriques

$$G_\lambda(x, y) = \begin{pmatrix} 1 + (I_x^\lambda)^2 & I_x^\lambda I_y^\lambda \\ I_x^\lambda I_y^\lambda & 1 + (I_y^\lambda)^2 \end{pmatrix} \quad (100)$$

On peut enfin traiter  $I$  comme une variété  $M$  de dimension 3 de  $\mathbb{R}^4$  paramétrée par

$$\begin{aligned} \psi : \Omega \times \Omega' &\longrightarrow \mathbb{R}^4 \\ \psi(x, y, \lambda) &= (x, y, \lambda, I(x, y, \lambda) = I^\lambda(x, y)) \end{aligned} \quad (101)$$

où  $\Omega'$  est le domaine spectral de l'image. En un point  $p$  de  $M$ , l'espace tangent  $T_p M$  est de dimension 3 avec pour base les vecteurs

$$\psi_x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ I_x \end{pmatrix} \quad \psi_y = \begin{pmatrix} 0 \\ 1 \\ 0 \\ I_y \end{pmatrix} \quad \psi_\lambda = \begin{pmatrix} 0 \\ 0 \\ 1 \\ I_\lambda \end{pmatrix} \quad (102)$$

Dans ce dernier cas, la métrique de  $M$  est une matrice de  $SPD(3)$  :

$$G(x, y, \lambda) = \begin{pmatrix} 1 + I_x^2 & I_x I_y & I_x I_\lambda \\ I_x I_y & 1 + I_y^2 & I_y I_\lambda \\ I_x I_\lambda & I_y I_\lambda & 1 + I_\lambda^2 \end{pmatrix} \quad (103)$$

Dans la pratique, nous allons travailler avec une matrice quelque peu différente et ce pour des raisons d'implémentation de la dérivée par rapport à la longueur d'onde. La matrice, notée désormais  $G(x, y)$ , que nous considérons est donnée par

$$G(x, y) = \begin{pmatrix} 1 + \sum_{\lambda=1}^{\Lambda} (I_x^\lambda)^2 & \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_y^\lambda & \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_t^\lambda \\ \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_y^\lambda & 1 + \sum_{\lambda=1}^{\Lambda} (I_y^\lambda)^2 & \sum_{\lambda=1}^{\Lambda} I_y^\lambda I_t^\lambda \\ \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_t^\lambda & \sum_{\lambda=1}^{\Lambda} I_y^\lambda I_t^\lambda & 1 + \sum_{\lambda=1}^{\Lambda} (I_t^\lambda)^2 \end{pmatrix} \quad (104)$$

où la dérivée partielle  $I_t^\lambda$  est obtenue comme suit :

$$I_t^\lambda = (I^{\lambda-1} * \mathfrak{G}) - (I^{\lambda+1} * \mathfrak{G}) \quad (105)$$

où  $*$  est le produit de convolution et  $\mathfrak{G}$  est un masque  $3 \times 3$ . Le masque  $\mathfrak{G}$  choisi pour les test est

$$\mathfrak{G} = \begin{pmatrix} 0.065 & 0.065 & 0.065 \\ 0.065 & 0.480 & 0.065 \\ 0.065 & 0.065 & 0.065 \end{pmatrix} \quad (106)$$

**Remarque 6.** La matrice  $G(x, y)$  peut être considérée comme la première forme fondamentale de la variété de dimension 3 paramétrée par

$$\begin{aligned} \tilde{\psi} : \Omega \times \Omega' &\longrightarrow \mathbb{R}^{3+\Lambda} \\ \tilde{\psi}(x, y, t) &= (x, y, t, I^1(x, y, t), I^2(x, y, t), \dots, I^\Lambda(x, y, t)) \end{aligned} \quad (107)$$

en introduisant un "temps virtuel"  $t$ .

Les énoncés classiques de la géométrie riemannienne montrent que la donnée de l'ensemble des matrices (98), (100) ou (104), permet d'obtenir toutes les informations concernant la géométrie dite intrinsèque des variétés correspondantes. On dispose de formules explicites pour notamment la courbure de Gauss des surfaces ou l'équation différentielle d'ordre deux qui caractérisent les géodésiques (voir [20] ou [69] par exemple). Nous avons vu rapidement dans la section 3.1.3 comment exploiter la métrique riemannienne d'une surface  $S$  pour obtenir des détecteurs de contours pour une image en niveau de gris.

Le but de ce chapitre est d'explorer la possibilité de mettre en place des algorithmes de classification pour la segmentation non supervisée d'images hyperspectrales à partir de la seule prise en compte de l'ensemble des matrices (98), (100) et (104) de  $SPD(2)$  ou  $SPD(3)$ . Nous allons tirer profit du fait que l'ensemble  $SPD(d)$  peut être muni d'une métrique riemannienne pertinente pour mesurer les distances entre ces matrices.

## 4.2 La géométrie de l'espace $SPD(d)$

Rappelons que  $SPD(d)$  désigne l'ensemble des matrices  $d \times d$  à coefficients réels symétriques et définies positives. La géométrie de l'espace  $SPD(d)$  se décrypte de la même façon que la géométrie du demi-plan de Poincaré. C'est ce qu'a mis en évidence Siegel dans [66]. Nous commençons donc par un court rappel sur ce demi-plan (voir par exemple [70]).

### 4.2.1 $SPD(1)$ et le demi-plan de Poincaré

L'espace  $SPD(1)$  n'est autre que l'ensemble des réels strictement positifs. Le demi-plan de Poincaré est l'ensemble

$$\mathbb{H} = \{x \in \mathbb{C}, x = x_1 + ix_2, x_2 > 0\} \quad (108)$$

Évidemment  $SPD(1)$  s'identifie (via la multiplication par  $i$ ) aux points de  $\mathbb{H}$  de partie réelle nulle. La métrique de Poincaré est définie sur  $\mathbb{H}$  par

$$ds_P^2(dx) = \frac{dx_1^2 + dx_2^2}{x_2^2} \quad (109)$$

et correspond à la donnée en tout point  $x = (x_1, x_2)$  de  $\mathbb{H}$  de la matrice symétrique définie positive

$$G_P(x_1, x_2) = \begin{pmatrix} 1/x_2^2 & 0 \\ 0 & 1/x_2^2 \end{pmatrix} \quad (110)$$

Si l'on ne considère que les points de partie réelle nulle, on obtient en restreignant cette métrique

$$ds_P^2(idx_2) = dx_2^2/x_2^2 \quad (111)$$

ce qui permet d'obtenir une métrique riemannienne sur  $SPD(1)$

$$\boxed{ds_{SPD(1)}^2(dx_2) = dx_2^2/x_2^2} \quad (112)$$

Voyons rapidement comment calculer la distance entre deux réels positifs  $a$  et  $b$  avec  $b > a$ . Soit

$$\gamma : t \mapsto (1-t)a + tb \quad (113)$$



où  $t \in [0, 1]$  le segment reliant  $ia$  à  $ib$  dans  $\mathbb{H}$ . Le vecteur tangent en  $\gamma(t)$  au segment  $\gamma$  a pour coordonnées  $(0, b - a)$ . On a donc

$$l(\gamma) = \int_0^1 \frac{b - a}{a + t(b - a)} dt \quad (114)$$

D'où  $l(\gamma) = \log(b/a)$ . Cette longueur est la distance géodésique entre les deux éléments  $a$  et  $b$  de  $SPD(1)$ . Plus généralement,

$$\boxed{d_{SPD(1)}(a, b) = |\log a - \log b|} \quad (115)$$

Il existe de nombreuses justifications du bien fondé de cette métrique sur  $\mathbb{H}$ . D'une part, elle possède un groupe d'isométrie très pertinent du point de vue géométrique, le groupe  $PSL(2, \mathbb{R})$ . Ce groupe est l'ensemble des matrices  $2 \times 2$  à coefficients réels de déterminant 1 quotienté par  $\{-Id, Id\}$ . Par ailleurs elle coïncide à un facteur près avec la métrique de Fisher des gaussiennes univariées (voir [1]). En effet, ces dernières ont pour densité les fonctions

$$p(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x - \mu)^2}{2\sigma^2}\right] \quad (116)$$

et les coefficients de la métrique de Fisher sont donnés par

$$[G_F(\mu, \sigma)]_{11} = \int_{-\infty}^{+\infty} \left[ \frac{\partial \log p(x, \mu, \sigma)}{\partial \mu} \frac{\partial \log p(x, \mu, \sigma)}{\partial \mu} \right] p(x, \mu, \sigma) dx \quad (117)$$

$$[G_F(\mu, \sigma)]_{22} = \int_{-\infty}^{+\infty} \left[ \frac{\partial \log p(x, \mu, \sigma)}{\partial \sigma} \frac{\partial \log p(x, \mu, \sigma)}{\partial \sigma} \right] p(x, \mu, \sigma) dx \quad (118)$$

$$[G_F(\mu, \sigma)]_{12} = [G_F(\mu, \sigma)]_{21} = \int_{-\infty}^{+\infty} \left[ \frac{\partial \log p(x, \mu, \sigma)}{\partial \mu} \frac{\partial \log p(x, \mu, \sigma)}{\partial \sigma} \right] p(x, \mu, \sigma) dx \quad (119)$$

Il est facile de vérifier que

$$ds_F^2(dp(x, \mu, \sigma)) = \frac{d\mu^2 + 2d\sigma^2}{\sigma^2} \quad (120)$$

La métrique riemannienne de  $SPD(1)$  coïncide donc au facteur 2 près avec la métrique de Fisher des gaussiennes univariées centrées ( $\mu = 0$ ). Rappelons enfin que pour la métrique de Poincaré, le demi-plan  $\mathbb{H}$  est de courbure de Gauss constante négative égale à  $-1$ .

### 4.2.2 $SPD(d)$ et le demi-espace de Siegel

Le demi-espace de Siegel  $\mathbb{S}(d)$  est l'ensemble

$$\mathbb{S}(2) = \{X = X_1 + iX_2, X_2 > 0\} \quad (121)$$

des matrices  $d \times d$ ,  $X = X_1 + iX_2$ , symétriques à coefficients complexes dont la partie imaginaire

$$X_2 = \frac{1}{2i}(X - \bar{X}) \quad (122)$$

appartient à  $SPD(d)$ , ce que l'on note  $X_2 > 0$ . L'ensemble  $SPD(d)$  s'identifie donc aux éléments de  $\mathbb{S}(d)$  de partie réelle nulle. L'espace  $\mathbb{S}(d)$  est muni d'une action transitive du



groupe symplectique  $Sp(2d, \mathbb{R})$  (qui se substitue au groupe  $PSL(2, \mathbb{R})$  de  $\mathbb{H}$ ). Siegel montre dans [66] que la forme hessienne du birapport

$$R(X, Y) = (X - Y)(X - \bar{Y})^{-1}(\bar{X} - \bar{Y})(\bar{X} - Y)^{-1} \quad (123)$$

est invariante sous l'action de  $Sp(2d, \mathbb{R})$  et est définie positive. Il s'agit de la forme hermitienne

$$ds_{\mathbb{S}}^2(dX) = \text{trace}(X_2^{-1}dX X_2^{-1}d\bar{X}) \quad (124)$$

En restreignant comme précédemment cette forme hermitienne aux éléments de  $\mathbb{S}(d)$  de partie réelle nulle, on obtient la définition suivante.

**Définition 11.** *La métrique de Rao sur l'ensemble  $SPD(d)$  des matrices  $d \times d$  à coefficients réels définies positives est la métrique de Siegel*

$$ds_{\mathbb{S}}^2(dX) = \text{trace}(X_2^{-1}dX X_2^{-1}d\bar{X}) \quad (125)$$

restreinte aux éléments de  $\mathbb{S}(d)$  de partie réelle nulle. Elle est donnée par

$$ds_{SPD(d)}^2(dX_2) = \text{trace}(X_2^{-1}dX_2 X_2^{-1}dX_2) = \text{trace}(X_2^{-1}dX_2)^2 \quad (126)$$

On peut vérifier comme ci-dessus que cette métrique correspond à la métrique de Fisher de l'espace des gaussiennes multivariées centrées. En utilisant la norme de Frobénius

$$\|A\|_F^2 = \langle A, A \rangle_F \quad (127)$$

où

$$\langle A, B \rangle_F = \text{trace}(AB^t) \quad (128)$$

on obtient une formule simple (à comparer à la formule (112))

$$\boxed{ds_{SPD(d)}^2(dX_2) = \|X_2^{-1}dX_2\|_F^2} \quad (129)$$

Le calcul de la distance géodésique, la distance de Siegel, se fait en utilisant l'action doublement transitive du groupe symplectique  $Sp(2d, \mathbb{R})$ . En se restreignant aux éléments de  $\mathbb{S}(d)$  de partie réelle nulle, on obtient la définition suivante.

**Définition 12.** *La distance de Rao sur l'ensemble  $SPD(d)$  des matrices  $d \times d$  à coefficients réels définies positives est définie par*

$$\boxed{d_{SPD(d)}^2(X_2, Y_2) = \|\log(X_2^{-1/2}Y_2X_2^{-1/2})\|_F^2 = \|\log(X_2^{-1}Y_2)\|_F^2} \quad (130)$$

Dans cette dernière formule  $\log$  désigne le logarithme matriciel. Dans le paragraphe suivant nous aurons besoin de la paramétrisation explicite de la géodésique reliant deux matrices symétriques définies positives  $X_2$  et  $Y_2$ .

**Proposition 2.** *La géodésique  $\gamma$  de  $SPD(d)$  reliant deux éléments  $X_2 = \gamma(0)$  et  $Y_2 = \gamma(1)$  est la courbe de  $SPD(d)$  paramétrée par*

$$\boxed{\gamma : s \in [0, 1] \mapsto X_2^{1/2}(X_2^{-1/2}Y_2X_2^{-1/2})^s X_2^{1/2}} \quad (131)$$

Pour la suite de ce chapitre, nous adoptons les notations suivantes. Les éléments de  $SPD(d)$  sont désignés par des lettres majuscules ( $X, M, A, \dots$ ) sans indice et la distance de Rao est notée  $d_R$ ,

$$d_R(M, N) = \|\log(M^{-1}N)\|_F \quad (132)$$

### 4.3 Moyennes et médianes dans $SPD(d)$

Comme nous l'avons déjà dit, le but de ce chapitre est de mettre en place un algorithme de  $k$ -means dans l'espace  $SPD(d)$  des matrices  $d \times d$  à coefficients réels définies positives. Pour cela, il est nécessaire de tout d'abord définir une bonne notion de moyenne.

#### 4.3.1 Barycentre de Karcher

On trouve déjà en 1929 la remarque suivante de Cartan (voir [14], paragraphe 16) : “*Or dans un espace de Riemann sans point singulier à distance finie, simplement connexe, à courbure négative ou nulle, on peut trouver, étant donnés des points en nombre fini, un point fixe invariant par tous les déplacements qui échangent entre eux les points donnés : c'est le point pour lequel la somme des carrés des distances au point donné est minima*”. La définition que l'on adopte est la suivante.

**Définition 13.** Soit  $M$  une variété riemannienne et soit  $d_M$  la distance géodésique sur  $M$ . Soient  $p_1, p_2, \dots, p_n$ ,  $n$  points de  $M$ , un barycentre de Karcher de ces points est définie par

$$\bar{p} = \operatorname{argmin}_{p \in M} \sum_{i=1}^n d_M^2(p, p_i) \quad (133)$$

Nous reviendrons sur le problème de l'existence et de l'unicité d'un tel barycentre dans le contexte qui est le nôtre. Nous renvoyons à [34] pour l'histoire quelque peu curieuse des différentes appellations de cette moyenne (barycentre riemannien, centre de masse riemannien, moyenne de Fréchet...). Avant toute chose, et comme nous l'avons fait dans le paragraphe précédent, nous allons traiter le cas simple et instructif de  $SPD(1) = \mathbb{R}_+^*$ . On vérifie facilement que si  $d_E$  désigne la distance euclidienne  $d_E^2(x, y) = (x - y)^2$  sur  $\mathbb{R}_+^*$  alors le point  $\bar{x}$  tel que

$$\bar{x} = \operatorname{argmin}_{x \in \mathbb{R}_+^*} \sum_{i=1}^n (x - x_i)^2 \quad (134)$$

n'est autre que la moyenne arithmétique des  $x_i$ ,  $i = 1, 2, \dots, n$ . Voyons quelle moyenne on obtient en utilisant la métrique issue de la géométrie de  $\mathbb{H}$ .

**Proposition 3.** La moyenne de Karcher sur  $SPD(1) = \mathbb{R}_+^*$  de  $n$  réels strictement positifs  $x_1, x_2, \dots, x_n$  pour la distance  $d_{SPD(1)}$

$$d_{SPD(1)}(x, y) = |\log x - \log y| \quad (135)$$

est la moyenne géométrique

$$\bar{x} = (x_1 x_2 \cdots x_n)^{1/n} \quad (136)$$

des  $x_i$ ,  $i = 1, \dots, n$ .

Il s'agit donc de minimiser la fonction

$$\varphi(x) = \sum_{i=1}^n (\log x - \log x_i)^2 = \sum_{i=1}^n (\log(x_i^{-1}x))^2 \quad (137)$$

Pour cela nous allons calculer son gradient pour la métrique  $ds_{SPD(1)}^2$ . On a par définition

$$\langle \nabla_x \varphi, u \rangle_{SPD(1)} = d_x \varphi(u) \quad (138)$$

pour tout  $u$  de l'espace tangent  $T_x(SPD(1)) \simeq \mathbb{R}$  à  $SPD(1)$  en  $x$ . On a donc

$$\frac{\nabla_x \varphi}{x^2} = \frac{2}{x} \sum_{i=1}^n (\log x - \log x_i) \quad (139)$$

soit

$$\boxed{\nabla_x \varphi = 2x \sum_{i=1}^n \log(x_i^{-1}x)} \quad (140)$$

Comme  $x > 0$  et  $x_i > 0$ ,  $i = 1, 2, \dots, n$ , alors

$$\nabla_x \varphi = 0 \iff \sum_{i=1}^n \log(x_i^{-1}x) = 0 \iff \log\left(\frac{x^n}{x_1 x_2 \cdots x_n}\right) = 0 \quad (141)$$

et par suite

$$\bar{x} = (x_1 x_2 \cdots x_n)^{1/n} \quad (142)$$

est la moyenne géométrique des  $x_i$ ,  $i = 1, 2, \dots, n$ .

La démarche est tout à fait similaire dans le cas de l'espace  $SPD(d)$ . Nous allons en particulier obtenir une formule analogue pour le gradient. Cependant, les calculs sont un peu plus compliqués. Le résultat suivant est plus ou moins bien connu (voir [53]).

**Théorème 1.** *La moyenne de Karcher sur  $SPD(d)$  de  $n$  matrices  $X_1, X_2, \dots, X_n$  symétriques définies positives pour la distance*

$$d_R(M, N) = \|\log(M^{-1}N)\|_F \quad (143)$$

existe et est l'unique solution de l'équation

$$\sum_{i=1}^n \log(X_i^{-1}X) = 0 \quad (144)$$

Il s'agit une nouvelle fois de minimiser la fonction

$$\varphi(X) = \sum_{i=1}^n \|\log(X_i^{-1}X)\|_F^2 \quad (145)$$

Tout comme le demi-plan de Poincaré  $\mathbb{H}$ , l'espace riemannien  $SPD(d)$  est à courbure négative (voir [67]) ce qui permet d'assurer l'existence et l'unicité de la moyenne de Karcher dans  $SPD(d)$  (voir [35]). Pour obtenir la caractérisation donné par le théorème, on procède comme précédemment. On a

$$\langle \nabla_X \varphi, U \rangle_{SPD(d)} = d_X \varphi(U) \quad (146)$$

ou encore

$$\text{trace}(X^{-1} \nabla_X \varphi X^{-1} U) = d_X \varphi(U) \quad (147)$$

pour toute matrice symétrique  $U$  de l'espace tangent  $T_X SPD(d)$ . La difficulté ici est d'obtenir l'expression de la différentielle  $d_X \varphi$ . Nous proposons un calcul un peu différent de celui effectué dans [53]. Notons

$$F(X) = \|\log(AX)\|_F^2 \quad (148)$$

avec  $A$  et  $X$  dans  $SPD(d)$ . Alors

$$d_X F(U) = 2\langle F(X), d_{AX} \log(AU) \rangle_F \quad (149)$$

Or (voir [19], formule (3.13))

$$d_{AX} \log(AU) = \int_0^1 ((AX - I)t + I)^{-1} AU ((AX - I)t + I)^{-1} dt \quad (150)$$

D'où

$$d_X F(U) = 2\text{trace} \left[ \log(AX) \int_0^1 ((AX - I)t + I)^{-1} AU ((AX - I)t + I)^{-1} dt \right] \quad (151)$$

et par suite,

$$d_X F(U) = 2\text{trace} \left[ \log(AX) \int_0^1 ((AX - I)t + I)^{-2} dt AU \right] \quad (152)$$

Comme

$$\int_0^1 ((AX - I)t + I)^{-2} dt = (I - AX)^{-1} [(AX)^{-1} - I] = (AX)^{-1} \quad (153)$$

alors

$$d_X F(U) = 2\text{trace} [\log(AX)(AX)^{-1} AU] = 2\langle \log(AX), X^{-1}U \rangle_F \quad (154)$$

On obtient donc

$$d_X \varphi(U) = 2 \sum_{i=1}^n \langle \log(X_i^{-1}X), X^{-1}U \rangle_F \quad (155)$$

d'où, d'après (147),

$$\boxed{\nabla_X \varphi = 2X \sum_{i=1}^n \log(X_i^{-1}X)} \quad (156)$$

Cette dernière formule est à comparer avec la formule (140). Comme  $\varphi$  est une somme de fonctions convexes,

$$\bar{X} = \underset{X \in SPD(d)}{\operatorname{argmin}} \varphi(X) \iff \nabla_{\bar{X}} \varphi = 2\bar{X} \sum_{i=1}^n \log(X_i^{-1}\bar{X}) = 0 \iff \sum_{i=1}^n \log(X_i^{-1}\bar{X}) = 0 \quad (157)$$

### 4.3.2 Barycentre itératif

Il est bien connu que si  $\mathbb{X}_k$  est une suite de variables aléatoires i.i.d. sur  $\mathbb{R}^n$ , la moyenne

$$m_k = \frac{\mathbb{X}_1 + \mathbb{X}_2 + \cdots + \mathbb{X}_k}{k} = \frac{k-1}{k}m_{k-1} + \frac{1}{k}\mathbb{X}_k \quad (158)$$

converge presque partout vers l'espérance  $\mathbb{E}[\mathbb{X}_1]$ . La construction du barycentre itératif s'inspire de ce résultat classique en adaptant la moyenne (158) et en exploitant une nouvelle fois la géométrie à courbure négative de  $SPD(d)$  (voir [71]). Soient  $n$  matrices  $X_1, X_2, \dots, X_n$  de  $SPD(d)$  et soit

$$(\Omega, P) = \prod_{k=1}^{\infty} (\Omega_k, P_k) \quad (159)$$

l'espace probabilisé tel que  $\Omega_k = (\xi_1, \xi_2, \dots, \xi_n)$  et

$$P_k = \sum_{i=1}^n \delta_{\xi_i} \quad (160)$$

pour tout  $k$ ,  $\delta_{\xi_i}$  désignant la mesure de Dirac en  $\xi_i$ . On considère la suite de variables aléatoires  $\mathbb{X}_k$  à valeurs dans  $SPD(d)$  définies par : pour tout  $\omega$  de  $\Omega$ ,

$$\mathbb{X}_k(\omega) = X_i \quad (161)$$

si la  $k$ ième composante de  $\omega$  est  $\xi_i$ . Ces variables  $\mathbb{X}_k$  sont i.i.d. avec comme loi de probabilité

$$q = \sum_{i=1}^n \delta_{X_i} \quad (162)$$

Le barycentre de la mesure  $q$  est défini par (voir [71])

$$\bar{q} = \operatorname{argmin}_{X \in SPD(d)} \int_{SPD(d)} [d_R^2(X, Z) - d_R^2(Y, Z)] q(dZ) \quad (163)$$

soit

$$\bar{q} = \operatorname{argmin}_{X \in SPD(d)} \sum_{i=1}^n [d_R^2(X, X_i) - d_R^2(Y, X_i)] \quad (164)$$

et est caractérisé par la condition (proposition 4.3 de [71])

$$\boxed{\bar{q} = \operatorname{argmin}_{X \in SPD(d)} \sum_{i=1}^n d_R^2(X, X_i)} \quad (165)$$

Ainsi le barycentre de la mesure  $q$ ,  $\bar{q}$ , coïncide-t-il avec le barycentre de Karcher  $\bar{X}$  introduit précédemment.

On a de plus,

$$\mathbb{E}[\mathbb{X}_1] = \operatorname{argmin}_{X \in SPD(d)} \int_{\Omega} [d_R^2(X, \mathbb{X}_1(\omega)) - d_R^2(Y, \mathbb{X}_1(\omega))] P(d\omega) = \bar{q} \quad (166)$$

Voyons maintenant comment adapter la loi des grands nombres. Nous avons déjà vu que l'unique géodésique reliant deux matrices  $X$  et  $Y$  de  $SPD(d)$  est la courbe paramétrée par

$$s \in [0, 1] \mapsto X^{1/2}(X^{-1/2}YX^{-1/2})^s X^{1/2} \quad (167)$$

Nous la notons

$$X \sharp_s Y \quad (168)$$

On considère la suite de variables aléatoires

$$\mathbb{S}_1 = \mathbb{X}_1, \quad \mathbb{S}_k = \mathbb{S}_{k-1} \sharp_{1/k} \mathbb{X}_k \quad (169)$$

avec les variables aléatoires  $\mathbb{X}_k$  définies par (161). On a alors le théorème suivant (voir par exemple [30]).

**Théorème 2.** *La suite  $\mathbb{S}_k$  converge presque partout vers l'espérance  $\mathbb{E}[\mathbb{X}_1] = \bar{q}$ .*

Finalement, en considérant la suite des moyennes

$$\boxed{S_k = S_{k-1} \sharp_{1/k} X_k} \quad (170)$$

avec  $S_1 = X_1$ , on obtient une très bonne approximation du barycentre de Karcher. L'avantage de ce procédé itératif est qu'il est très peu coûteux en temps de calcul.

### 4.3.3 Médiane

Le but ici est de décrire comment obtenir une médiane pour un ensemble de matrices de  $SPD(d)$ . La démarche est sensiblement la même que pour la barycentre de Karcher.

**Définition 14.** *Soient  $X_1, X_2, \dots, X_n$  des matrices de  $SPD(d)$ , une médiane de cet ensemble de matrices est définie par*

$$\boxed{\tilde{X} = \operatorname{argmin}_{X \in SPD(d)} \sum_{i=1}^n \|X - X_i\|_F} \quad (171)$$

Il s'agit donc de minimiser la fonction

$$\psi(X) = \sum_{i=1}^n \|X - X_i\|_F \quad (172)$$

différentiable sur  $SPD(d) \setminus \{X_1, X_2, \dots, X_n\}$ . On a

$$D_X \psi(h) = \sum_{i=1}^n \frac{\langle X - X_i, h \rangle_F}{\|X - X_i\|_F} \quad (173)$$

pour  $X \neq X_i$ ,  $1 \leq i \leq n$ . D'où

$$\operatorname{trace}(X^{-1}(\nabla_X \psi)X^{-1}h) = \sum_{i=1}^n \frac{\operatorname{trace}((X - X_i)^t h)}{\|X - X_i\|_F} \quad (174)$$

soit encore

$$\text{trace}(X^{-1}(\nabla_X \psi)X^{-1}h) = \text{trace} \left[ \sum_{i=1}^n \left( \frac{(X - X_i)^t}{\|X - X_i\|_F} \right) h \right] \quad (175)$$

et par suite

$$\text{trace}(X^{-1}(\nabla_X \psi)X^{-1}h) = \text{trace} \left[ X^{-1} \sum_{i=1}^n X \left( \frac{(X - X_i)^t}{\|X - X_i\|_F} \right) X X^{-1} h \right] \quad (176)$$

Finalement,

$$\nabla_X \psi = \sum_{i=1}^n \left[ X \left( \frac{(X - X_i)^t}{\|X - X_i\|_F} \right) X \right] = X \left[ \sum_{i=1}^n \left( \frac{(X - X_i)^t}{\|X - X_i\|_F} \right) \right] X \quad (177)$$

Ce gradient utilise la structure riemannienne de  $SPD(d)$ . Il n'est valable que si  $X \neq X_i$ ,  $1 \leq i \leq n$ . On a donc

$$\nabla_X \psi = 0 \iff \sum_{i=1}^n \frac{(X - X_i)}{\|X - X_i\|_F} = 0 \quad (178)$$

soit encore

$$\tilde{X} = \left( \sum_{i=1}^n \frac{X_i}{\|\tilde{X} - X_i\|_F} \right) / \left( \sum_{i=1}^n \frac{1}{\|\tilde{X} - X_i\|_F} \right) \quad (179)$$

pour  $X \neq X_i$ ,  $1 \leq i \leq n$ .

Il existe un moyen simple pour vérifier si l'une des matrices  $X_i$  est la médiane cherchée (voir [39]). On a, pour  $t > 0$ ,

$$\psi(X_k + th) = \sum_{i=1, i \neq k}^n \|X_k + th - X_i\|_F + \|X_k + th - X_k\|_f \quad (180)$$

soit

$$\psi(X_k + th) = \sum_{i=1, i \neq k}^n \|X_k + th - X_i\|_F + t\|h\|_F \quad (181)$$

D'où

$$\lim_{t \rightarrow 0} \frac{\psi(X_k + th) - \psi(X_k)}{t} = \sum_{i=1, i \neq k}^n \frac{\langle X_k - X_i, h \rangle_F}{\|X_k - X_i\|_F} + \|h\|_F \quad (182)$$

Fixons la norme  $\|h\|_F$  de  $h$  à 1 et notons

$$R_k = \sum_{i=1, i \neq k}^n \frac{X_k - X_i}{\|X_k - X_i\|_F} \quad (183)$$

La dérivée (182) est la plus petite possible quand  $h = -R_k/\|R_k\|_F$ ; elle vaut alors

$$\lim_{t \rightarrow 0} \frac{\psi(X_k + th) - \psi(X_k)}{t} = -\|R_k\|_F + 1 \quad (184)$$

Ainsi

$$\left\| \sum_{i=1, i \neq k}^n \frac{X_k - X_i}{\|X_k - X_i\|_F} \right\|_F \leq 1 \quad (185)$$

si et seulement si  $X_k$  est la médiane des  $X_i$  pour  $i = 1, 2, \dots, n$ . Cette caractérisation permet de pallier en amont le problème de la non différentiabilité de la fonction  $\psi$  en les  $X_i$ .

## 4.4 Classification et segmentation

La figure 29 illustre schématiquement la méthode de segmentation que nous proposons de mettre en œuvre dans ce chapitre consacré à la géométrie riemannienne. Nous revenons tout d'abord sur l'implémentation des calculs de barycentres de Karcher. Ensuite nous présentons des résultats de clustering dans  $SPD(2)$ ,  $SPD(2)^\Lambda$  et  $SPD(3)$ .

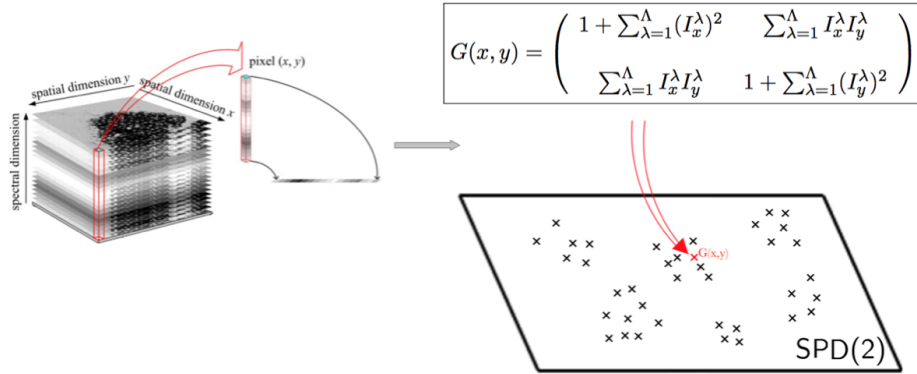


FIGURE 29 – Représentation schématique de la segmentation par clustering dans  $SPD(2)$

Nous l'avons vu, il y a essentiellement deux façons d'aborder le problème du calcul du barycentre de Karcher d'un ensemble de matrices de  $SPD(d)$  : en utilisant une descente de gradient à l'aide de (156) (voir par exemple [6] ou [53]) ou en utilisant un algorithme récursif à l'aide de (170) (voir par exemple [30] ou [51]). Nous comparons rapidement ces deux possibilités pour justifier du bien fondé de notre choix.

Soit  $X$  un ensemble de  $n$  matrices  $X_1, X_2, \dots, X_n$  de  $SPD(d)$ . Notons

$$V_X(\overline{X}_\nabla) = \sum_{i=1}^n d_R^2(X_i, \overline{X}_\nabla) \quad (186)$$

où  $\overline{X}_\nabla$  désigne le barycentre de Karcher des  $X_i$  obtenu à partir d'une descente de gradient, et

$$V_X(\overline{X}_\#) = \sum_{i=1}^n d_R^2(X_i, \overline{X}_\#) \quad (187)$$

où  $\overline{X}_\#$  désigne le barycentre de Karcher des  $X_i$  obtenu à partir de l'algorithme récursif. Pour évaluer les performances de chacun des deux algorithmes, nous avons généré 900 échantillons



$X^j$ ,  $1 \leq j \leq 900$ , chaque échantillon  $X^j$  contenant  $j$  matrices de  $SPD(2)$ . La figure 30(a)

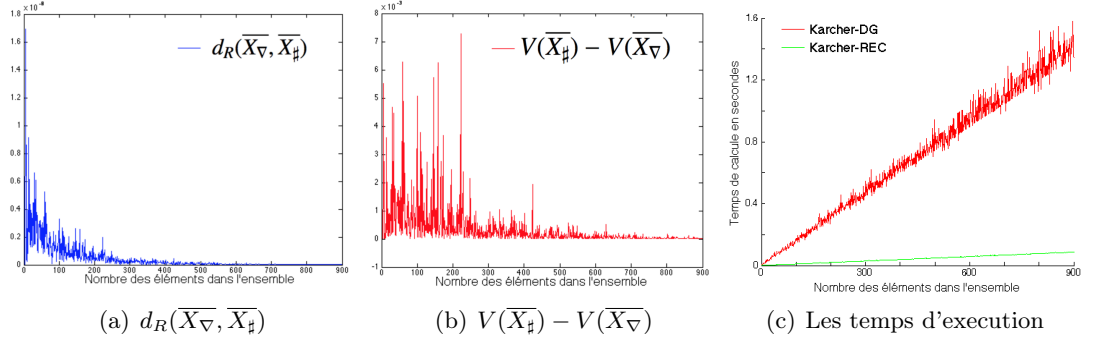


FIGURE 30 – Comparaison entre les deux algorithmes de calcul de moyennes de Karcher.

montre que la distance de Rao entre  $\overline{X}_{\nabla}$  et  $\overline{X}_{\#}$  est toujours très petite et quasi nulle dès que l'échantillon contient plus de 500 matrices. La figure 30(b) montre la différence entre les valeurs  $V(\overline{X}_{\#})$  et  $V(\overline{X}_{\nabla})$ . On constate que cette différence est toujours positive, ce qui signifie que l'algorithme par descente de gradient est en quelque sorte plus performant que l'algorithme récursif. Cependant, cette différence tend rapidement vers 0 quand l'échantillon contient plus de 600 matrices. La figure 30(c) montre très clairement l'avantage qu'il y a à choisir l'algorithme récursif compte tenu du très grand nombre de matrices que nous avons à considérer pour nos applications.

#### 4.4.1 Clustering dans $SPD(2)$

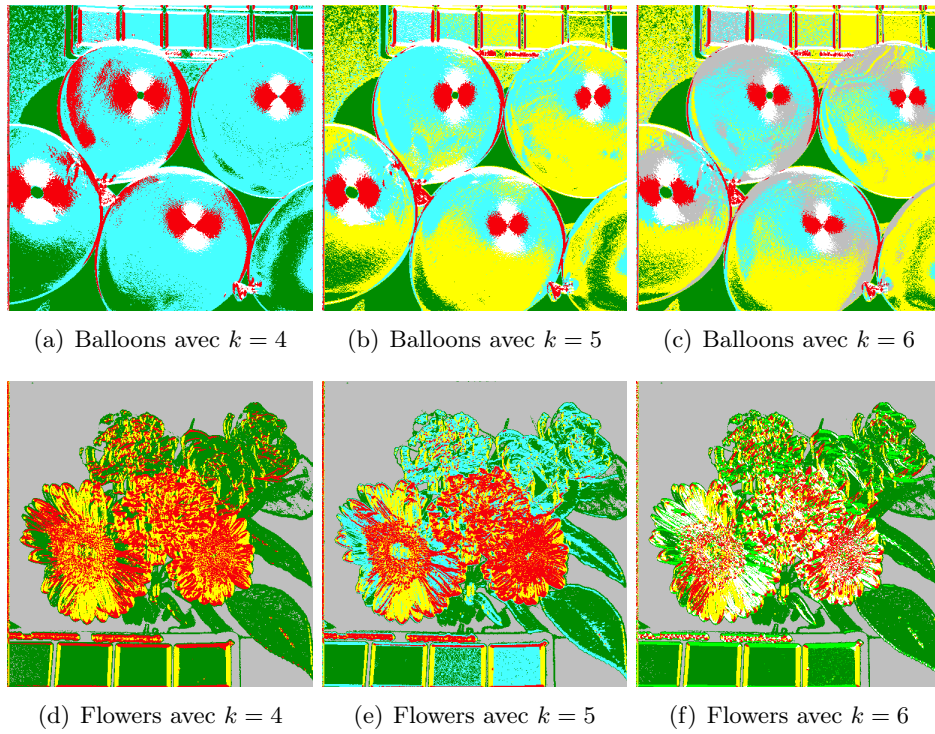
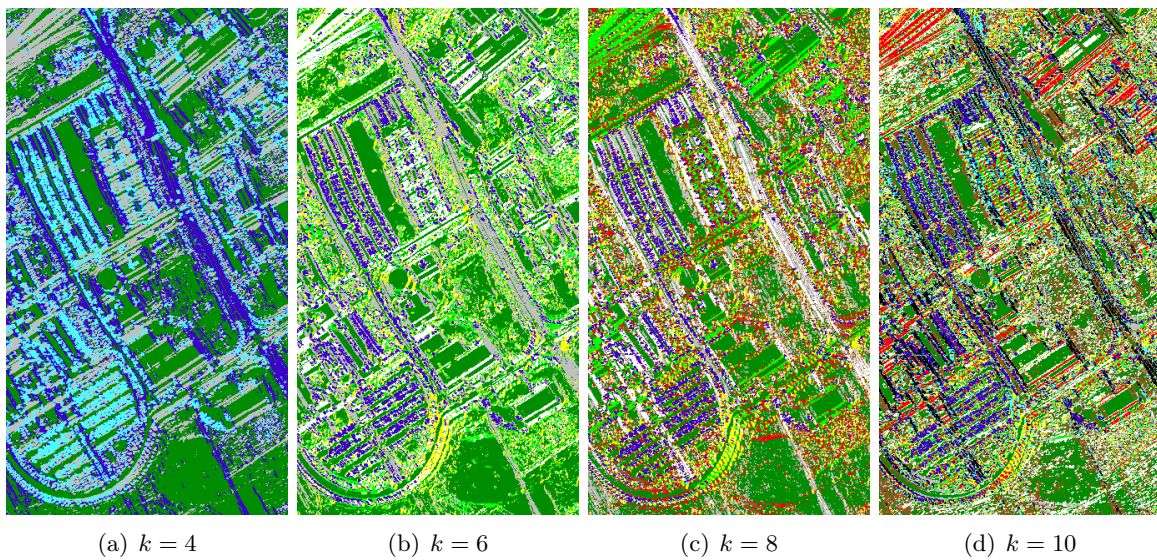
Les résultats présentés dans cette partie sont obtenus en appliquant l'algorithme des  $k$ -means dans  $SPD(2)$ , c'est-à-dire en considérant les matrices

$$G(x, y) = \begin{pmatrix} 1 + \sum_{\lambda=1}^{\Lambda} (I_x^{\lambda})^2 & \sum_{\lambda=1}^{\Lambda} I_x^{\lambda} I_y^{\lambda} \\ \sum_{\lambda=1}^{\Lambda} I_x^{\lambda} I_y^{\lambda} & 1 + \sum_{\lambda=1}^{\Lambda} (I_y^{\lambda})^2 \end{pmatrix} \quad (188)$$

et en utilisant l'algorithme récursif pour le calcul des barycentres de Karcher. Les images test sont "Flowers" et "Balloons" ([41]) de taille  $512 \times 512 \times 31$  et "PaviaU" ([76]) de taille  $340 \times 610 \times 103$ . Les nombres de classes sont 4, 5 et 6 pour "Flowers" et "Balloons" et 4, 6, 8 et 10 pour "PaviaU".

Ces premiers résultats montrent que l'algorithme des  $k$ -means dans  $SPD(2)$  est sensible au bruit (voir par exemple les figures 32(c) et 32(d)). Pour tenter de remédier à ce problème, il est possible d'utiliser des filtres médians en utilisant les résultats mentionnés dans la section 4.3.3. Pour cela, on utilise une descente de gradient (à l'aide de l'équation (177)) en prenant soin d'effectuer au préalable le test décrit par la condition (185). Les figures 33 et 34 montrent les nouveaux résultats que l'obtient (le nombre des classes est le même que précédemment).

La figure 35 permet de voir sur des parties des images "Balloons", avec  $k = 6$ , "Flowers", avec  $k = 5$ , et "PaviaU", avec  $k = 10$ , les effets des filtres médians. On constate que ces échantillons sont effectivement moins bruités.

FIGURE 31 – Clustering dans  $SPD(2)$  des images Balloons et Flowers.FIGURE 32 – Clustering dans  $SPD(2)$  de l'image PaviaU.

#### 4.4.2 Clustering dans $SPD(2)^\Lambda$

On s'intéresse ici à la segmentation en considérant comme feature space l'espace  $SPD(2)^\Lambda$ , c'est-à-dire en utilisant les familles de matrices

$$G_\lambda(x, y) = \begin{pmatrix} 1 + (I_x^\lambda)^2 & I_x^\lambda I_y^\lambda \\ I_x^\lambda I_y^\lambda & 1 + (I_y^\lambda)^2 \end{pmatrix} \quad (189)$$

pour  $1 \leq \lambda \leq \Lambda$ . Pour  $G^1(x, y) = (G_1^1(x, y), \dots, G_\Lambda^1(x, y))$  et  $G^2(x, y) = (G_1^2(x, y), \dots, G_\Lambda^2(x, y))$  deux telles familles, on définit

$$d_R^\Lambda(G^1(x, y), G^2(x, y)) = \left( \sum_{\lambda=1}^{\Lambda} d_R^2(G_\lambda^1, G_\lambda^2) \right)^{1/2} \quad (190)$$

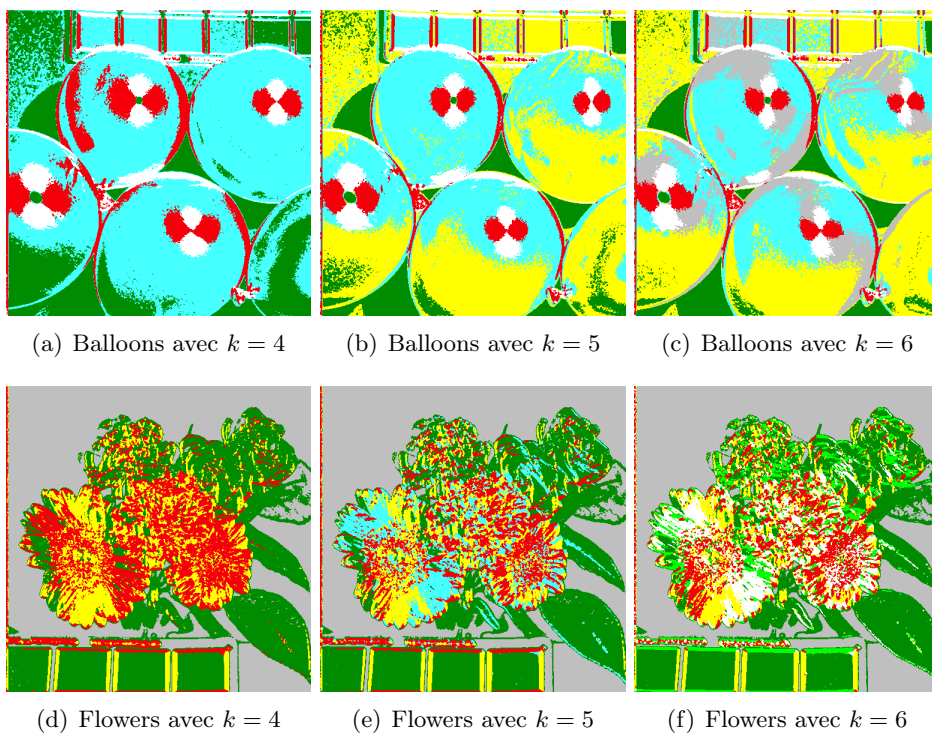
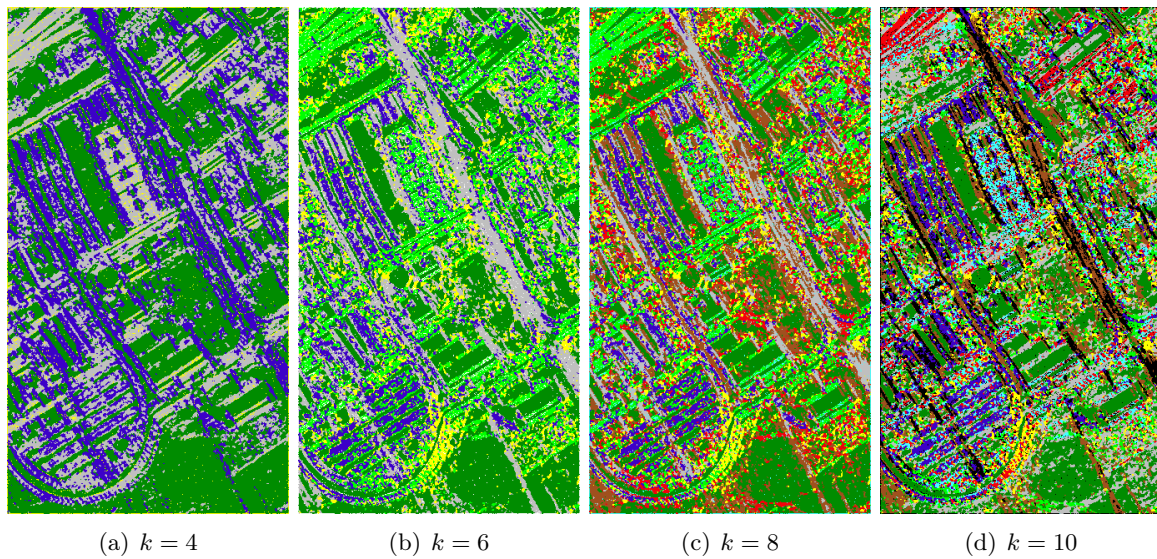
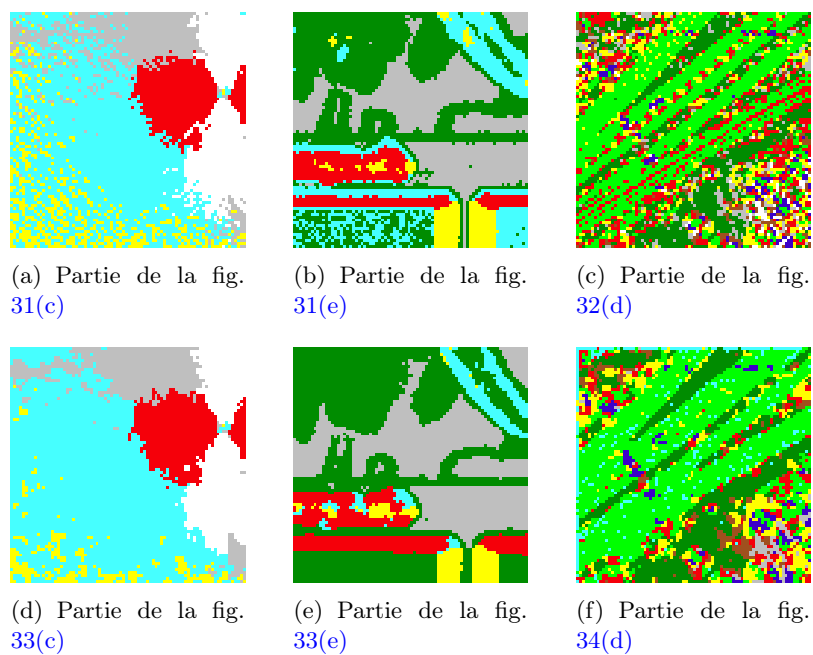


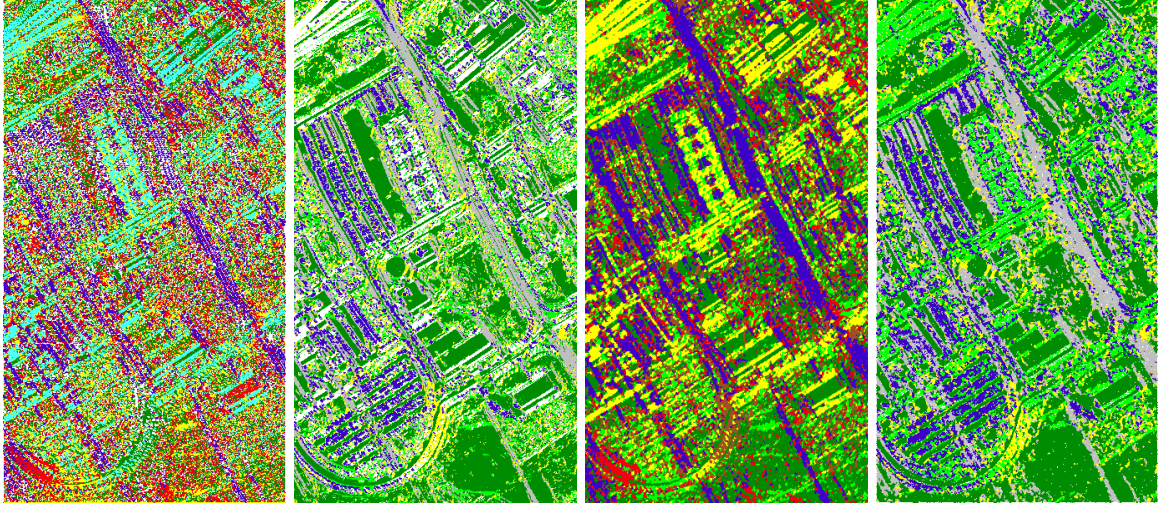
FIGURE 33 – Clustering avec médiane dans  $SPD(2)$  des images Balloons et Flowers.



FIGURE 34 – Clustering avec médiane dans  $SPD(2)$  de l'image PaviaU.FIGURE 35 – Les effets des filtres médians dans  $SPD(2)$ .

L'algorithme de  $k$ -means est modifié en conséquence, les centres étant actualisés composante par composante. On utilise une nouvelle fois l'algorithme récursif pour déterminer les barycentres de Karcher.

Cette deuxième approche est évidemment plus coûteuse en temps de calcul. La figure 36



(a)  $SPD(2)^\Lambda$  sans filtres medians (b)  $SPD(2)$  sans filtres medians (c)  $SPD(2)^\Lambda$  avec filtres medians (d)  $SPD(2)$  avec filtres medians

FIGURE 36 – Comparaisons des résultats de clustering dans  $SPD(2)^\Lambda$  et  $SPD(2)$  sur l'image PaviaU avec  $k = 6$ .

montre les résultats que l'obtient sur l'image "PaviaU" avec  $k = 6$  en comparaison avec les résultats obtenus dans  $SPD(2)$ . On constate que cette nouvelle approche ne donne pas de meilleurs résultats et semble même plus sensible au bruit que l'approche précédente.

#### 4.4.3 Clustering dans $SPD(3)$

Nous nous intéressons finalement à la segmentation par clustering dans  $SPD(3)$ , c'est-à-dire à partir des matrices

$$G(x, y) = \begin{pmatrix} 1 + \sum_{\lambda=1}^{\Lambda} (I_x^\lambda)^2 & \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_y^\lambda & \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_t^\lambda \\ \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_y^\lambda & 1 + \sum_{\lambda=1}^{\Lambda} (I_y^\lambda)^2 & \sum_{\lambda=1}^{\Lambda} I_y^\lambda I_t^\lambda \\ \sum_{\lambda=1}^{\Lambda} I_x^\lambda I_t^\lambda & \sum_{\lambda=1}^{\Lambda} I_y^\lambda I_t^\lambda & 1 + \sum_{\lambda=1}^{\Lambda} (I_t^\lambda)^2 \end{pmatrix} \quad (191)$$

et en utilisant l'algorithme récursif pour le calcul des barycentres de Karcher.

Les figures 37 et 38 montrent les résultats obtenus pour les images "Balloons" et "Flowers" avec  $k = 4, 5, 6$  et pour l'image "PaviaU" avec  $k = 4, 6, 8$ . Une comparaison entre les trois méthodes de clustering (dans  $SPD(2)$ , dans  $SPD(2)^\Lambda$  et dans  $SPD(3)$ ) est présentée en figure 39. Le clustering dans  $SPD(3)$  semble donner des résultats un peu meilleurs que ceux obtenus par le clustering dans  $SPD(2)$ .

**Remarque 7.** A l'instar de la méthode de clustering par codage binaire du chapitre 3 et contrairement aux méthodes de clustering dans  $SPD(2)$  ou  $SPD(2)^\Lambda$ , la méthode de clustering dans  $SPD(3)$  tient compte de l'ordre des bandes spectrales. Une permutation de ces dernières doit donc influencer les résultats. Pour le vérifier, on choisit, au hasard, la permutation

$$\sigma = (2 \ 10)(5 \ 15)(8 \ 17)(20 \ 100)(24 \ 25)(30 \ 37) \quad (192)$$

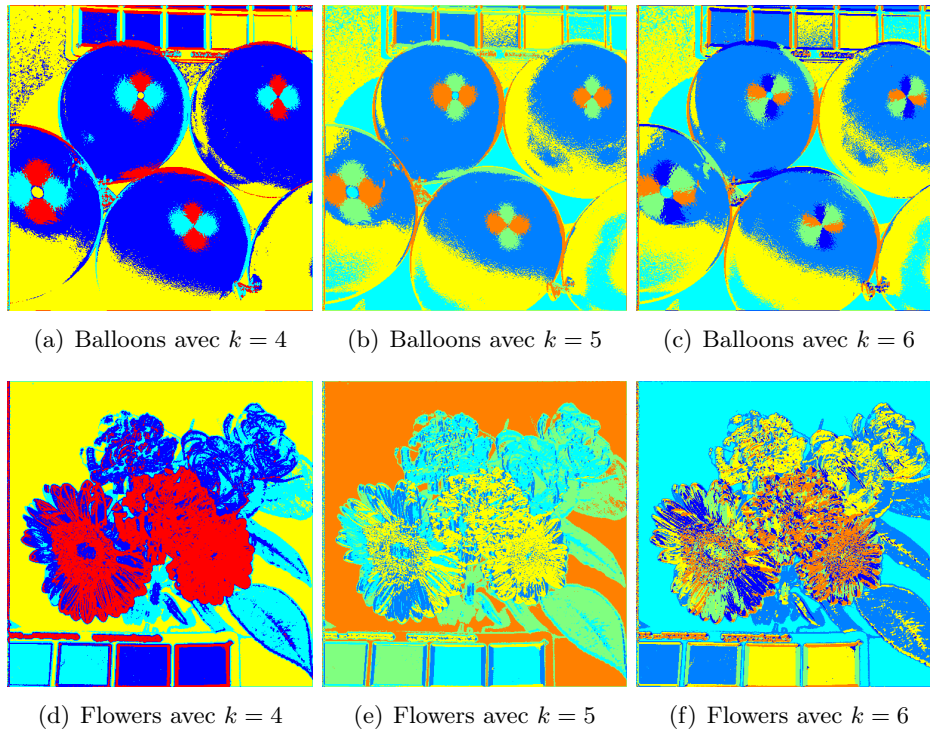


FIGURE 37 – Clustering dans  $SPD(3)$  des images Balloons et Flowers.

La figure 40 montre les résultats de segmentation obtenus sans et après permutation des bandes spectrales. Il est surprenant de constater que les résultats après application de la permutation  $\sigma$  semblent (bien) meilleurs que les résultats initiaux sans permutation. Nous nous limitons ici à cette constatation, l'explication et l'exploitation de ce phénomène faisant partie de nos perspectives.

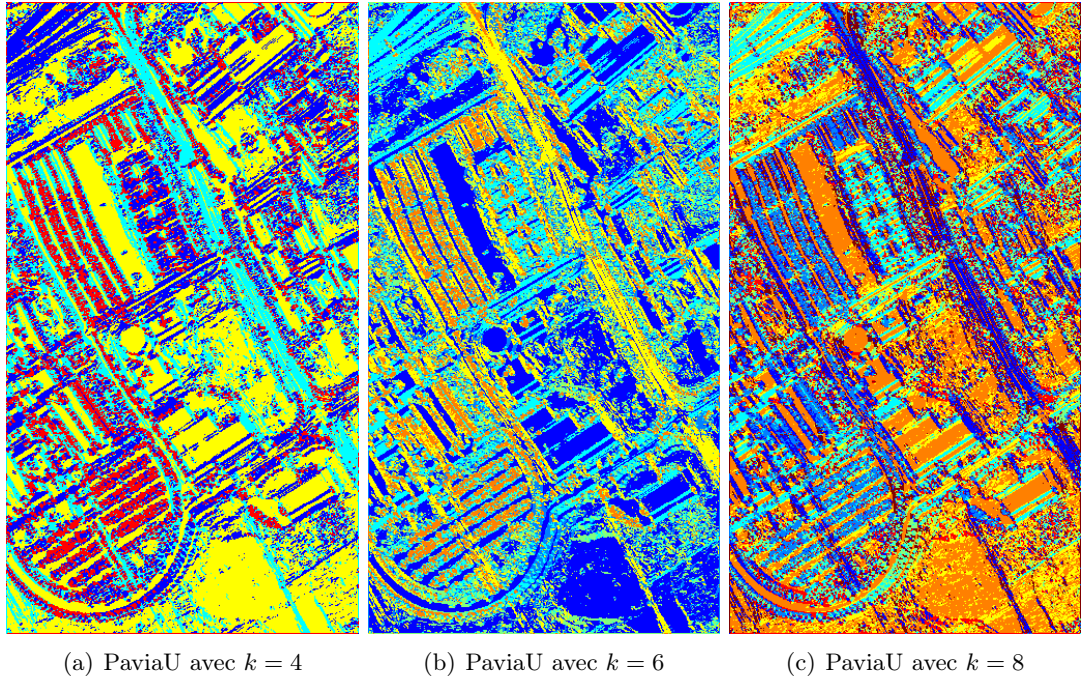
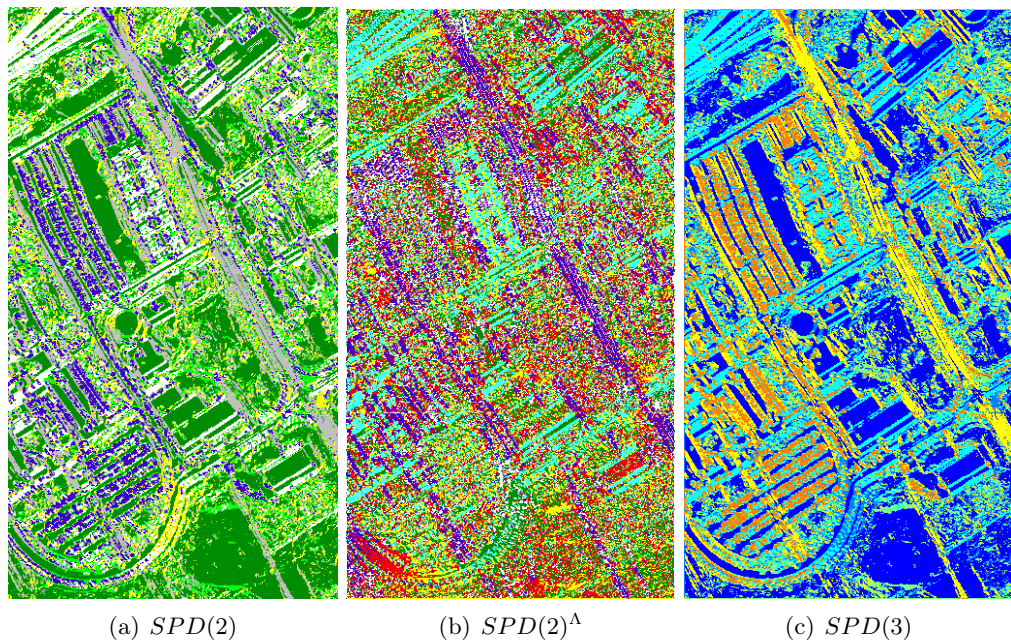
## 4.5 Conclusions et perspectives du chapitre

Le but de chapitre était d'explorer les possibilités d'utiliser les outils de la géométrie riemannienne pour la segmentation non supervisée d'images hyperspectrales. Cette étude est en partie motivée par le fait que très peu de travaux abordent l'analyse d'images hyperspectrales sous cet angle. Pour cela, nous avons considéré trois façons de représenter géométriquement une image hyperspectrale, comme une surface, comme une famille de surfaces et enfin comme une variété de dimension 3. Les algorithmes de segmentation obtenus reposent essentiellement sur les informations géométriques dont on dispose dans les espaces  $SPD(2)$  et  $SPD(3)$  grâce à la métrique de Rao.

Du point de vue algorithmique, on tire avantage de l'implémentation récursive du calcul des barycentres de Karcher.

Les premiers résultats montrent que les algorithmes de  $k$ -means dans  $SPD(2)$ ,  $SPD(2)^\Delta$  et  $SPD(3)$  sont très sensibles au bruit contenu dans les images et qu'il se révèle nécessaire de



FIGURE 38 – Clustering dans  $SPD(3)$  de l'image PaviaU.FIGURE 39 – Comparaison des trois clusterings sans filtres médians sur l'image PaviaU avec  $k = 6$ .

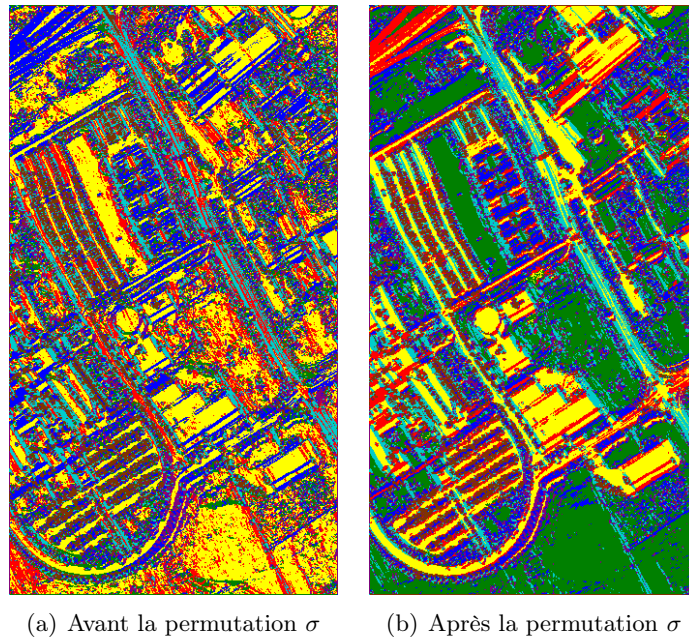


FIGURE 40 – Clustering dans  $SPD(3)$  de l'image PaviaU sans et avec la permutation spectrale  
– Les paramètres d'initialisation sont identiques et  $k = 7$ .

procéder à un pré-traitement des données. Nous avons choisi un filtre médian dans  $SPD(d)$  utilisant la norme de Frobenius, voir (171). Un choix plus judicieux mais plus compliqué à mettre en pratique serait d'utiliser un filtre médian défini à partir de la distance de Rao. Par ailleurs, d'autres choix de distances sont possibles sur  $SPD(d)$ , comme la distance de Stein (voir [64]), et il fait partie de nos perspectives de comparer les algorithmes issus de ces choix.

Finalement, il semblerait que travailler dans l'espace  $SPD(2)^\Lambda$  n'apporte pas grand chose compte tenu de "l'accumulation" des bruits de chacune des surfaces de la famille  $S_\lambda$ ,  $1 \leq \lambda \leq \Lambda$ . Par contre, le phénomène décrit dans la remarque 7 est intrigant et doit être étudié plus attentivement.





# Chapitre 5

## Conclusion

### 5.1 Résumé des travaux de la thèse

Nous nous sommes intéressés dans ce travail au problème de la segmentation non supervisée d’images hyperspectrales suivant trois approches bien distinctes.

La première, de type *Graph Embedding*, nécessite deux étapes :

- une étape d’appariement des pixels de patches de l’image initiale grâce à une mesure de similarité spectrale entre pixels ;
- une étape d’appariement d’objets issus des segmentations locales grâce à une mesure de similarité entre objets.

La deuxième, de type *Spectral Hashing* ou *Semantic Hashing*, repose sur un codage binaire des variations des profils spectraux. On procède à des segmentations par clustering à l’aide d’un algorithme de  $k$ -modes adapté au caractère binaire des données à traiter et à l’aide d’une version généralisée de la distance classique de Hamming.

La troisième utilise les informations riemanniennes des graphes (variétés) issus des différentes façons de représenter géométriquement une image hyperspectrale. Les segmentations se font une nouvelle fois par clustering à l’aide d’un algorithme de  $k$ -means. Nous exploitons pour cela les propriétés géométriques de l’espace des métriques, ou de façon équivalente de l’espace des matrices symétriques définies positives, induites par la métrique de Fisher Rao.

Ces trois approches envisagent des espaces de caractéristiques, *feature spaces*, très différents : l’espace  $[0, 1]^\Lambda$ , l’espace  $\{0, 1\}^{\Lambda-1}$  et l’espace  $SPD(d)$  (avec  $d = 2$  ou  $d = 3$ ). Du point de vue algorithmique, la première se distingue des deux autres. Alors que les approches par codage binaire ou riemannienne profitent de l’efficacité des algorithmes  $k$ -modes et  $k$ -means, la première approche, par similarité, nécessite de comparer exhaustivement les pixels des patches puis les objets issus des segmentations locales et de procéder à des fermetures transitives de graphes.

### 5.2 Comparaisons des trois approches

La première remarque que l’on peut faire concerne effectivement ces espaces de caractéristiques. On se convainc assez facilement qu’ils sont de “tailles” très différentes :  $[0, 1]^\Lambda$  est un cube d’un espace vectoriel de dimension  $\Lambda$ ,  $\{0, 1\}^{\Lambda-1}$  est un ensemble fini de

cardinal  $2^{\Lambda-1}$  et  $SPD(2)$  (resp.  $SPD(3)$ ) est un cône de l'espace vectoriel  $\mathbb{R}^3$  (resp.  $\mathbb{R}^6$ )<sup>1</sup>. Comme nous l'avons déjà signalé, l'approche par codage binaire repose sur une réduction très conséquente de la taille des données. Il faut également souligner que l'espace des caractéristiques de l'approche riemannienne est le même quelque soit le nombre  $\Lambda$  de longueurs d'onde considérées. C'est en partie cette propriété qui a motivé l'étude menée dans le chapitre 4 et c'est en partie aussi ce qui en fait son intérêt.

La deuxième remarque porte sur la complexité des algorithmes mis en œuvre. Comme mentionné plus, l'approche par similarité nécessite des comparaisons exhaustive qui sont très coûteuses en temps de calcul, c'est assurément son point faible. Le partitionnement du chapitre 3 repose sur l'algorithme des  $k$ -modes qui requiert le calcul de distances de Hamming généralisées entre un nombre relativement petit de codes binaires. Il est donc bien moins coûteux en temps de calcul. Quant à l'approche riemannienne du chapitre 4, l'algorithme de  $k$ -means nécessite le calcul d'un nombre assez restreint de barycentres de Karcher. Nous avons déjà insisté sur les performances de la méthode itérative décrite dans la section 4.3.2 pour obtenir ces barycentres.

La dernière remarque a trait à la qualité des résultats obtenus. Comme on peut facilement l'imaginer, la première approche donne les meilleurs résultats. Les segmentations de l'approche par codage binaire sont tout à fait comparables avec celles référencées dans la littérature traitant des méthodes non supervisées. Les premiers résultats obtenus dans le chapitre 4 montrent qu'il est nécessaire d'effectuer un prétraitement afin de débruiter les données. Ceci est dû au fait que la méthode s'appuie sur des considérations géométriques impliquant des calculs de dérivées. On peut cependant noter la qualité des résultats obtenus par clustering dans  $SPD(3)$  sans filtres médians.

Cette discussion est résumée dans le tableau 5.

Approche	Taille de l'espace	Complexité algorithmique	Résultats
Similarité	Grande	Grande	Bons
Codages binaires	Petite	Faible	Satisfaisants
Riemannienne	Moyenne et constante	Faible	Corrects

TABLE 5 – Tableau de comparaisons entre les trois approches proposées.

### 5.3 Perspectives

Nous avons déjà eu l'occasion d'évoquer quelques-unes de nos perspectives de travail en fin de chapitres et en particulier les possibilités d'étendre certaines de nos approches avec des outils issus de la logique floue. Nous nous proposons de tester ici une version floue de l'algorithme des  $k$ -modes [32] pour la classification des codes binaires du chapitre 3.

Soit donc  $I$  une image hyperspectrale de taille spatiale  $M \times N$  et de taille spectrale  $\Lambda$ . Rappelons que pour chaque pixel  $(m, n)$ , le codage binaire fournit un vecteur  $I_b(m, n)$  de dimension  $p = \Lambda - 1$  qui est un objet à classifier de l'ensemble

$$X = \{I_b(m, n) \in \{0, 1\}^p, (m, n) \in E\} = \{x_i \in \{0, 1\}^p, 1 \leq i \leq M \times N\} \quad (193)$$

1. On ne considère pas ici le clustering dans  $SPD(2)^\Lambda$  car il ne nous a pas semblé convaincant.

de cardinal  $M \times N$ . La nouvelle fonctionnelle à minimiser, qui remplace la fonctionnelle (79), est

$$J_{DHG}^\beta(U, C) = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^\beta DHG(x_i, c_j) \quad (194)$$

où  $u_{ij} \in [0, 1]$  représente l'appartenance de l'objet  $x_i$  au cluster  $j$  associé à son mode  $c_j$  et  $\beta$  est un paramètre réel strictement supérieur à 1. Un objet peut donc appartenir à plusieurs clusters et on a les contraintes naturelles

$$\sum_{j=1}^k u_{ij} = 1 \quad (195)$$

pour tout  $1 \leq i \leq n$  et

$$0 < \sum_{i=1}^n u_{ij} < n \quad (196)$$

pour tout  $1 \leq j \leq k$ .

Les étapes d'actualisation (80) et (81) sont modifiées en conséquence de la façon suivante.

1. Actualiser  $U^{(t)} : \forall 1 \leq i \leq n, \forall 1 \leq j \leq k$ ,

$$u_{ij}^{(t)} = \begin{cases} 1 & \text{si } x_i = c_j^{(t)} \\ 0 & \text{si } x_i = c_h^{(t)}, h \neq j \\ \left[ \sum_{m=1}^k \left( \frac{DHG(c_j^{(t)}, x_i)}{DHG(c_m^{(t)}, x_i)} \right)^{\frac{1}{\beta-1}} \right]^{-1} & \text{si } x_i \neq c_m^{(t)}, \forall 1 \leq m \leq k \end{cases} \quad (197)$$

2. Actualiser  $C^{(t)} : \forall 1 \leq j \leq k, \forall 1 \leq l \leq p$ ,

$$c_{jl}^{(t)} = \underset{a \in \{0,1\}}{\operatorname{argmin}} \|\tilde{c}_{jl}^{(t)} - x\| \quad (198)$$

avec

$$\tilde{c}_{jl}^{(t)} = \left[ \frac{\sum_{i=1}^n u_{ij}^\beta x_{il}}{\sum_{i=1}^n u_{ij}^\beta} \right] \quad (199)$$

L'actualisation (198) est à comparer avec l'actualisation (81) de l'algorithme non flou. On vérifie en effet facilement que

$$c_{jl}^{(t)} = \underset{a \in \{0,1\}}{\operatorname{argmin}} |\tilde{c}_{jl}^{(t)} - a| = 1 \iff 2 \times \sum_{i=1}^n u_{ij}^\beta x_{il} \geq \sum_{i=1}^n u_{ij}^\beta \quad (200)$$

et par suite

$$c_{jl}^{(t)} = 1 \iff \sum_{i=1}^n u_{ij}^\beta x_{il} \geq \sum_{i=1}^n u_{ij}^\beta (1 - x_{il}) \quad (201)$$

avec  $c_{jl}^{(t)} = 1$  si  $\tilde{c}_{jl}^{(t)} = 1/2$ . La figure 41 montre les résultats que l'on obtient pour l'image "PaviaU" avec différentes valeurs du paramètre  $\beta$  et le codage binaire de la définition 7. Les meilleurs résultats sont obtenus pour les petites valeurs de  $\beta$  et font apparaître des objets

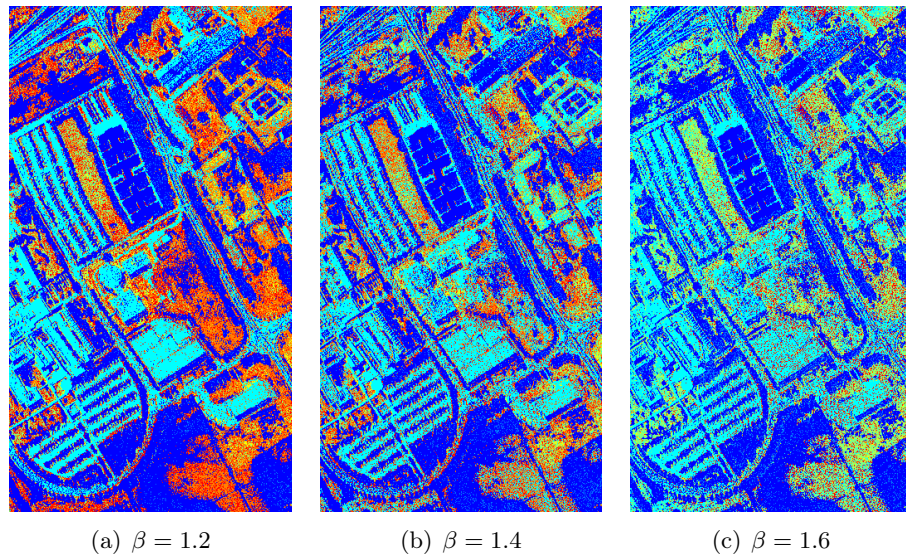


FIGURE 41 – Exemples de segmentation par l’algorithme de  $k$ -modes flou sur l’image PaviaU pour différentes valeurs du paramètre de flou  $\beta$ ,  $\alpha_S = 0.3$  et  $k = 7$ .

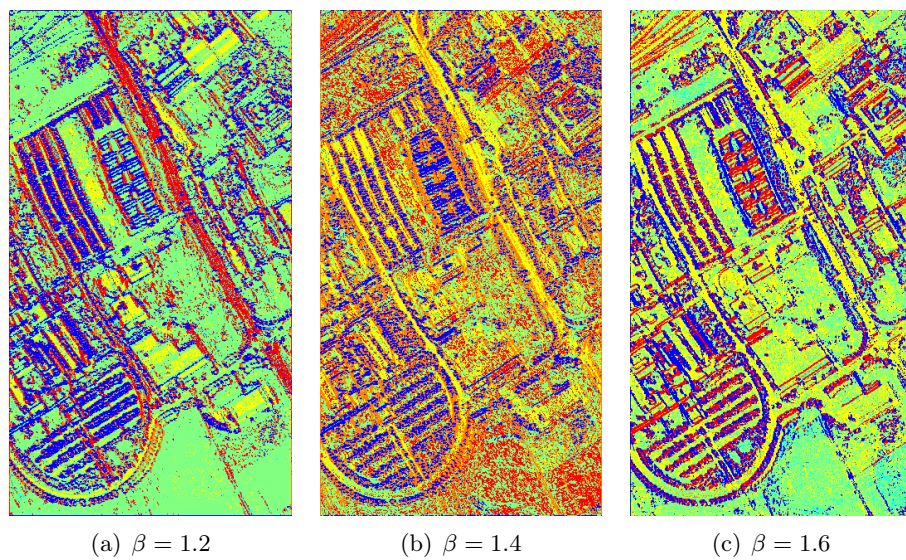


FIGURE 42 – Exemples de segmentation par l’algorithme de  $k$ -means flou dans  $SPD(3)$  sur l’image PaviaU pour différentes valeurs du paramètre de flou  $\beta$  et  $k = 7$ .

aux contours un peu mieux définis. Ces tests montrent que, pour notre problème de clustering de codes binaires, le fait de "fuzzifier" l'algorithme de  $k$ -modes n'apporte pas d'améliorations significatives au niveau des segmentations.

Il est possible de procéder de la même façon avec les algorithmes de classification dans  $SPD(2)$  et  $SPD(3)$  utilisés dans le chapitre 4. À titre d'exemple nous présentons dans la figure 42 des résultats obtenus à l'aide d'un algorithme de clustering  $k$ -means flou dans  $SPD(3)$ . Il faut remarquer que la formule d'actualisation des centres, qui est tout à fait semblable à la formule (199), ne fait plus intervenir les barycentres de Karcher. Par ailleurs, la convergence de l'algorithme qui utilise en entrée la distance géodésique de  $SPD(3)$  semble quelque peu erratique. Une analyse théorique du comportement de l'algorithme  $k$ -means flou avec cette distance reste peut-être à faire. Ces premiers tests montrent comme précédemment que les meilleurs résultats sont obtenus pour des petites valeurs du paramètre de flou  $\beta$  donc pour un clustering très proche du clustering non flou proposé dans le chapitre 4.



# Bibliographie

- [1] S.-I. Amari and H. Nagaoka. *Methods on Information Geometry*. American Mathematical Society, 2007.
- [2] G. Ball and B. Hall. Isodata, a novel method of data analysis and classification. Technical Report MTech. Rep. AD-699616, Stanford Univ., 1965.
- [3] S. Bandyopadhyay and S. Saha. *Unsupervised Classification : Similarity Measures, Classical and Metaheuristic Approaches, and Applications*. Springer-Verlag Berlin Heidelberg, 2013.
- [4] T. Batard, C. Saint Jean, and M. Berthier. A metric approach to nD images edge detection with Clifford algebras. *J Math Imaging Vis*, 33 :296–312, 2009.
- [5] J. Bezdek. *Pattern Recognition with fuzzy objective function algorithm*. Plenum Press, 1981.
- [6] D. A. Bini and B. Iannazzo. Computing the Karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4) :1700–1710, 2013.
- [7] D.A. Bini and B. Iannazzo. Toolbox Karcher Means.  
<http://bezout.dm.unipi.it/software/mtoolbox/mfiles/karcher.m>.
- [8] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gaber, and J. Chanussot. Hyperspectral unmixing overview : Geometrical, statistical, and sparse regression-based approaches. *IEEE J. Select. Topics Appl. Earth Observ. Remote Sensing*, 5(2) :354–379, 2012.
- [9] A. Bookstein, V. A. Kulyunkin, and T. Raita. Generalized Hamming distance. *Information Retrieval*, 5 :353–375, 2002.
- [10] G. Bougenière, C. Cariou, K. Chehdi, and A. Gay. Unsupervised non parametric data clustering by means of bayesian inference and information theory. In S.M.M. De Faria and Assuncao P.A.A., editors, *2th Int. Conf. on Signal Processing and Multimedia Applications*, pages 101–108. INSTICC Press, 2007.
- [11] G. Camp-Valls, D. Tuia, D. Gomez-Chova, S. Jimenez, and J. Malo. *Remote sensing image processing, synthesis lectures on image, video, and multimedia processing*. San Raphael, CA : Morgan and Claypool, 2011.
- [12] G. Camps-Valls, D. Tuia, L. Bruzzone, and A.J. Benediktsson. Advances in hyperspectral image classification. *IEEE Signal Processing Magazine*, 31(1) :45–54, 2014.
- [13] C. Cariou and K. Chehdi. Unsupervised nearest neighbors clustering with application to hyperspectral images. *IEEE J. of Select. Topics in Signal Processing*, 9(6) :1105–1116, 2015.



- [14] E. Cartan. Groupes simples clos et ouverts et géométrie riemannienne. *J. Math. Pures Appl.*, 8 :1–33, 1929.
- [15] G. Celeux and J. Diebolt. The sem algorithm : A probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Comput. Statist. Quart.*, 2 :73–82, 1985.
- [16] M.D. Craig. Unsupervised unmixing of remotely sensed images. In *Proc. Australian Remote Sensing Conf.*, pages 324–330, 1990.
- [17] M.M. Crawford, L. Ma, and W. Kim. Exploring nonlinear manifold learning in classification of hyperspectral data. In S. Prasad, L.M. Bruce, and J. Chanussot, editors, *Optical Remote Sensing : Advances in Signal Processing and Exploitation Techniques*, pages 207–234. London : Springer-Verlag, 2011.
- [18] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. of the Royal Stat. Soc., Series B*, 39(1) :1–38, 1977.
- [19] L. Dieci, B. Morini, and A. Papini. Computational techniques for real logarithms of matrices. *SIAM J. Matrix Anal. Appl.*, 17 :570–593, 1996.
- [20] M. Do Carmo. *Differential geometry of curves and surfaces*. Dover Publications ; 2 edition, 2016.
- [21] B.A. Dubrovin, A.T. Fomenko, and S.P. Novikov. *Modern Geometry Methods and Applications : Part I : The Geometry of Surfaces, Transformation Groups, and Fields*. Springer ; 2nd edition, 1991.
- [22] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. of Cybernetics*, 3(3) :32–57, 1971.
- [23] F. Emmert-Streib and M. Dehmer, editors. *Information Theory and Statistical Learning*. Springer US, 2009.
- [24] T. Frankel. *The Geometry of Physics : An Introduction*. Cambridge University Press ; 3 edition, 2011.
- [25] J.F. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814) :972–976, 2007.
- [26] K.S. Fu and J.K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1) :2–16, 1981.
- [27] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap. *Aggregation functions*. Cambridge University Press, 2009.
- [28] S. Helgason. *Differential geometry, Lie groups, and symmetric spaces*. American Mathematical Society, 2001.
- [29] S. Hesamoddin. Package K-Means.  
<http://www.cise.ufl.edu/~salehian/Softwares.html>.
- [30] J. Ho, G. Cheng, H. Salehian, and B. Vemuri. Recursive karcher expectation estimators and geometric law of large numbers. In *Proceedings of the Sixteenth international conference on artificial intelligence and statistics*, pages 325–332, 2013.
- [31] Z. Huang. Extensions to the  $k$ -means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2 :283–304, 1998.

- [32] Z. Huang and M.K. Ng. A fuzzy  $k$ -modes algorithm for clustering categorical data. *IEEE Trans. on Fuzzy Systems*, 7(4) :446–452, 1999.
- [33] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, 1988.
- [34] H. Karcher. Riemannian center of mass and so called Karcher mean. <http://arxiv.org/pdf/1407.2087v1.pdf>.
- [35] W. Kendall. Probability, convexity, and harmonic maps with small image : uniqueness and fine existence. *Proc. London Math. Soc.*, 6(2) :371–406, 1990.
- [36] N. Keshava and J.F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1) :44–57, 2002.
- [37] S.S.. Khan and A. Ahmad. Cluster center initialization algorithm for  $k$ -modes clustering. *Expert Systems with Applications*, 40 :7444–7456, 2013.
- [38] E. P. Klement, R. Mesiar, and E. Pap. *Triangular norms*. Springer, 2000.
- [39] H.W. Kuhn. A note on fermat’s problem. *Mathematical programming*, 4 :98–107, 1973.
- [40] H. Kwon and N.M. Nasrabadi. Kernel rx-algorithm : a nonlinear anomaly detector for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sensing*, 43(2) :388–397, 2005.
- [41] CAVE Laboratory. Curet database. <http://www1.cs.columbia.edu/CAVE/databases/multispectral/stuff/>.
- [42] T. Landgrebe. Hyperspectral image data analysis. *IEEE Signal Processing Magazine*, 19(1) :17–28, 2002.
- [43] H. Le Capitaine, T. Batard, C. Frélicot, and M. Berthier. Blockwise similarity in  $[0, 1]$  via triangular norms and Sugeno integrals - Application to cluster validity. *16th IEEE Int. Conf. on Fuzzy Systems*, 2007.
- [44] B. Long, Z. Zhang, and P. S. Yu. *Relational data clustering, models, algorithms, and applications*. Chapman and Hall/CRC, 2010.
- [45] D. Lungu, S. Prasad, M.M. Crawford, and O. Ersoy. Manifold-learning-based feature extraction for classification of hyperspectral data. *IEEE Signal Processing Magazine*, 31(1) :55–66, 2014.
- [46] W. Ma, J. Bioucas-Dias, J. Chanussot, and P.D. Gader. Signal and image processing in hyperspectral remote sensing. *IEEE Signal Processing Magazine*, 31(1), 2014.
- [47] D. Manolakis, M. Rossacci, D. Zhang, J. Cipar, R. Lockwood, T. Cooley, and J. Jacobson. Statistical characterization of hyperspectral background clutter in the reflective spectral region. *Appl. Opt.*, 47(28) :96–106, 2008.
- [48] D. Manolakis and G. Shaw. Detection algorithms for hyperspectral imaging applications. *IEEE Signal Processing Magazine*, 19(1) :29–43, 2002.
- [49] D. Manolakis, E. Truslow, M. Pieper, T. Cooley, and Brueggeman. Hyperspectral target detection. *IEEE Signal Processing Magazine*, 31(1) :24–33, 2014.
- [50] L. Mascarilla, M. Berthier, and C. Frélicot. A  $k$ -order fuzzy OR operator for pattern classification with  $k$ -order ambiguity rejection. *Fuzzy Sets and Systems*, 159(15) :2011–2029, 2008.
- [51] E. M. Massart, J. M. Hendrickx, and P.-A. Absil. Approximate matrix geometric means based on the inductive mean. <http://sites.uclouvain.be/absil/2015.09>, 2015.

- [52] S. Matteoli, M. Diani, and G. Corsini. A tutorial overview of anomaly detection in hyperspectral images. *IEEE Aerosp. Electron. Syst. Mag.*, 25(7) :5–27, 2010.
- [53] M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SAIM J. Matrix Anal. and Appl.*, 26(3) :735–747, 2005.
- [54] M. Nakahara. *Geometry, Topology and Physics*. CRC Press ; 2 edition, 2003.
- [55] N. M. Nasrabadi. Hyperspectral target detection : An overview of recent and future challenges. *IEEE Signal Processing Magazine*, 31(1) :34–44, 2014.
- [56] M. Nixon. *Feature Extraction and Image Processing*. Newnes, 2013.
- [57] C. Ordonez. Clustering binary data streams with k-means. In Halevy, A.Y et al., editor, *SIGMOD International Conference on Management of Data*, volume 338, pages 12–19. Ass. for Computing Machinery, 2003.
- [58] C. Ordonez. Clustering binary data streams with k-means. *DMKD '03 : Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 12–19, 2003.
- [59] C. M. Pieters and P. A. Englert. *Remote Geochemical Analysis ; Elemental and Mineralogic Composition*. Cambridge University Press, 1993.
- [60] J.B.M. Queen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [61] R. Quéré. *Quelques propositions pour la comparaison de partitions non strictes*. PhD thesis, Université de La Rochelle, 2012.
- [62] W. M. Rand. Objective criteria for the evaluation of clustering methods. *J. of the American Statistical Association*, 66, 1971.
- [63] I.S. Reed and X. Yu. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE Trans. Acoust., Speech, Signal Process.*, 38(10) :1760–1770, 1990.
- [64] H. Salehian, Guang Cheng, B.C. Vemuri, and J. Ho. Recursive estimation of the stein center of spd matrices and its applications. *2013 IEEE Int. Conf. on Computer Vision*, 1793-1800, 2013.
- [65] A.P. Schaum. Joint subspace detection of hyperspectral targets. In *Proc. IEEE Aerospace Conf. Big Sky, MT*, volume 3, pages 1818–1824, 2004.
- [66] C.L. Siegel. Symplectic geometry. *American Journal of Mathematics*, 65(1) :1–86, 1943.
- [67] L. Skovgaard. A riemannian geometry of the multivariate normal model. *Scand. J. Statistics*, 11 :211–223, 1984.
- [68] D.A. Socolinsky and L.B. Wolff. A new visualization paradigm for multispectral imagery and data fusion. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1. IEEE, 1999.
- [69] M. Spivak. *A comprehensive introduction to differential geometry*, volume Vol. 2. Publish or Perish ; 3rd edition, 1999.
- [70] S. Stahl. *The Poincaré half-plane : a gateway to modern geometry*. Jones and Bartlett Learning, 1993.

- [71] K.-T. Sturm. Probability measure on metric spaces of non positive curvature. In Auscher, P. et al., editor, *Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces*, volume 338. Contemp. Math., Amer. Math. Soc., 2003.
- [72] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition*, 43(7) :2367–2379, 2010.
- [73] A. Terras. *Harmonic analysis on symmetric spaces and applications*. New York : Springer-Verlag, 1988.
- [74] V. Vapnik. *Statistical Learning Theory*. New York : Wiley, 1998.
- [75] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. *Advances in Neural Information Processing Systems 21*, 1753-1760, 2009.
- [76] Wikipedia. Upv/ehu database. [http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral\\_Remote\\_Sensing\\_Scenes](http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes).