UNIVERSITÉ
DE REIMS
CHAMPAGNE-ARDENNE

Thèse de Doctorat

Pour l'obtention du titre de

Docteur de l'Université de Reims Champagne Ardenne
en Informatique

Visualisation et classification de données multidimensionnelles Application aux images multicomposantes

Frédéric Blanchard

Soutenue publiquement le 12 décembre 2005, à Reims.

Composition du jury :

Rapporteurs :
Madame Florence d'Alché, Professeur à l'Université d'Evry
Monsieur Stéphane Loiseau, Professeur à l'Université d'Angers
Examinateurs :
Madame Nozha Boujemaa, Directrice de Recherche à l'INRIA
Monsieur Francis Rousseaux, Professeur à l'Université de Reims
Directeur de thèse :
Monsieur Michel Herbin, Professeur à l'Université de Reims
Co-encadrement :
Monsieur Philippe Vautrot, Maître de Conférences à l'Université de Reims

A Chloé, Aude et mes parents

Résumé

L'analyse des images multicomposantes est un problème crucial. Les questions de la visualisation et de la classification de ces images sont très importantes. Nous nous sommes intéressés à ces problèmes en tentant de répondre aux questions : comment visualiser de façon immédiate et gobale une image multicomposante ? Et comment les informations sont elles structurées dans ces composantes ? Après nous être placés dans le cadre plus général de l'analyse des données et avant de nous intéresser aux problèmes de visualisation et de classification, nous nous sommes interrogés sur les problèmes liés à la dimensionnalité du problème et avons considéré le problème de la réduction de cette dimensionnalité. Nous avons ensuite apporté deux contributions dans les deux thématiques abordées. Nous avons tout d'abord mis au point une méthode de visualisation de données (et d'images multicomposantes) par l'image couleur. Notre approche statistique de la couleur en fait une méthode ne nécessitant aucune connaissance a priori sur les données. Composées d'une étape d'affectation des couleurs et d'une étape de spatialisation, elle permet de visualiser immédiatement les données en produisant une image résumant les principales caractéristiques de ces données. Nous avons testé cette méthode sur des données simulées et des bases de données réelles avec de très bon résultats. Nous proposons également une utilisation de notre méthode pour l'exploration dynamique de masses de données. Nous nous sommes intéressés ensuite au problème de la classification non supervisée. Notre contribution se situe plus exactement en amont de la classification proprement dite. Un nouveau système de représentation des données, et des liens entre les données, basé sur une transformation par rangs et l'utilisation de la théorie des ensembles flous a été conçu. Pour illustrer son efficacité, cette méthode de représentation à été utilisée dans un processus de classification avec d'excellents résultats. D'autres utilisations hors de ce contexte sont maintenant envisagées.
Nous avons au cours de tout ce travail apporté deux contributions aux problèmes qui nous intéressaient. Ces méthodes efficaces nous ont ouvert de nombreuses perspectives intéressantes et ont montré leur intérêt à l'aide de nombreux exemples.

Mots-clés

Visualisation, classification non supervisée, images multicomposantes, analyse de données, représentation des données, couleur, flou.

Remerciements

Je tiens à remercier Stéphane Loiseau et Florence d'Alché-Buc d'avoir accepté de rapporter ce travail de thèse. Je remercie également Nozha Boujemaa d'avoir bien voulu faire partie de mon jury, et Francis Rousseaux qui a présidé ce jury.

J'adresse également toute ma gratitude à mon directeur de thèse, Michel Herbin, pour m'avoir fait confiance, pour ses conseils et sa bienveillance, sa très grande disponibilité, ses idées à profusion et sa bonne humeur ; mon co-directeur de thèse Philippe Vautrot, pour sa gentillesse, son aide et ses avis éclairés ; mon tuteur pédagogique Stéphane Cormier ; Yannick Rémion et tous les membres du CReSTIC-LERI pour leur aide précieuse, leur soutien et pour la bonne humeur qui règne au laboratoire. Je remercie aussi particulièrement Aassif, qui compte tenu d'une heureuse simultanéité, a pu co-organiser les pots avec moi, et Jérôme, pour son aide et son soutien "logistique" sans faille et bénévole.

Naturellement, je remercie Aude pour son appui, sa présence et sa compréhension, et ma fille Chloé pour le bonheur quotidien qu'elle m'apporte et pour avoir bien voulu me laisser rédiger ce mémoire.
Je remercie également mes parents pour avoir fait de moi ce que je suis devenu et m'avoir soutenu pendant toutes ces années d'études.

Enfin, je salue chaleureusement mes amis : Matt, Fredop, Fab, Dam, Cab, Macq, Ben et Aurore, JC et Delphine, Greg et Dalen. Sans oublier Bayen Hélène et Alexandre, José Delphine Lucas et Cassy, Ros Laurence et Maé, Paulin et Gwendo, Didier et Jeanette ; et, pour terminer, les membres de ma famille dont je ne peux citer ici tous les noms pour des raisons de place.

SOMMAIRE

1 Introduction 1

2 Données multidimensionnelles 7

2.1 Les données multidimensionnelles . . . 9

2.1.1 Exemples de données multidimensionnelles . . . 9

2.1.2 Cas particulier des images multicomposantes . . . 13

2.2 Malédiction de la dimensionnalité . . . 14

2.2.1 Problème de représentation . . . 18

2.2.2 Phénomène d'espace vide (ou d'espace creux) . . . 18

2.2.2.1 Volume de la sphère unité . . . 18

2.2.2.2 Volumes de la sphère unité et du cube circonscrit . . . 19

2.2.2.3 Volumes de deux sphères . . . 19

2.2.2.4 Distributions Gaussiennes . . . 19

2.2.3 Phénomène de Hughes . . . 21

2.2.4 Hypothèse de normalité . . . 22

2.3 Dimension intrinsèque . . . 22

2.3.1 Méthodes locales . . . 23

2.3.2 Méthodes globales . . . 23

2.4 Méthodes de réduction de dimensionnalité . . . 25

2.4.1 Méthodes linéaires . . . 25

2.4.1.1 Analyse en Composantes Principales . . . 25

2.4.1.2 Poursuite de Projection . . . 32

2.4.1.3 Analyse en Composantes Indépendantes . . . 32

2.4.2 Méthodes non-linéaires . . . 34

2.4.2.1 Positionnement multidimensionnel . . . 34

2.4.2.2 Cartes de Kohonen . . . 34

2.4.2.3 Analyse en composantes curvilinéaires . . . 36

2.4.2.4 Autres méthodes . . . 36
2.5 Conclusion . . . 37

3 Visualisation **39**
3.1 La visualisation de données . . . 41
3.1.1 Méthodes de visualisation . . . 42
3.1.2 Quelques méthodes spécifiques de visualisation . . . 44
3.1.2.1 Matrices de Scatterplots . . . 44
3.1.2.2 Courbes d'Andrew . . . 45
3.1.2.3 Coordonnées parallèles . . . 47
3.1.2.4 Glyphes, icônes et métaphores . . . 49
3.1.2.5 Techniques Orientées-pixel . . . 53
3.1.3 Logiciels, Packages . . . 56
3.1.3.1 Logiciels et environnements de calculs numériques ou statistiques . . . 56
3.1.3.2 Logiciels dédiés à la visualisation . . . 56
3.1.4 Evaluation des techniques de visualisation . . . 57
3.2 Problématique et cadre de notre travail . . . 58
3.3 Visualisation par image couleur . . . 63
3.3.1 Réduction de dimensionnalité . . . 65
3.3.2 Calcul de la couleur d'une donnée de l'échantillon . . . 66
3.3.3 Construction d'une image . . . 68
3.3.4 Applications . . . 70
3.3.4.1 Visualisation des classes sur des données simulées . . . 71
3.3.4.2 Visualisation d'images multicomposantes . . . 78
3.3.4.3 Visualisation de bases de données multidimensionnelles réelles . . . 78
3.4 Visualisation dynamique . . . 81
3.4.1 Concept . . . 81
3.4.2 Application à la base de données IRIS . . . 84
3.5 Discussion et conclusion . . . 86

4 Classification **89**
4.1 Introduction . . . 90
4.2 Survol des méthodes de classification . . . 94
4.2.1 Méthodes hiérarchiques . . . 94
4.2.2 Méthodes de partitionnement . . . 98
4.2.2.1 Méthodes de réallocation . . . 99
4.2.2.2 Méthodes basées sur la densité de probabilité . . . 101
4.2.3 Méthodes de classification floue . . . 107

4.2.4 Conclusion 107
4.3 Une nouvelle représentation floue des données d'un échantillon 108
4.3.1 Exemple introductif 109
4.3.2 Représentation floue des données 112
4.3.2.1 Passage aux rangs 114
4.3.2.2 Les données comme sous-ensembles flous 114
4.3.2.3 L'échantillon comme sous-ensemble flou 116
4.3.2.4 Ensembles flous de connexion 120
4.4 Application à la classification 121
4.4.1 Principe 121
4.4.2 Exemples et applications 124
4.4.2.1 Données simulées 124
4.4.2.2 Image multicomposante 126
4.4.2.3 Données réelles 133
4.5 Discussion et conclusion 134

5 Conclusion **139**

Bibliographie. **156**

LISTE DES TABLES

2.1 Base de données *Iris de Fisher* 12
2.2 Paramètres d'acquisition des images CASI 13
2.3 Extrait du tableau de données correspondant aux images CASI de la figure 2.1 16
2.4 Extrait du tableau de données CASI 17
2.5 Pourcentage de variance portée par chacune des composantes principales . 30
2.6 Pourcentage cumulé de variance portée par chacune des composantes principales 30

3.1 Comparaison des techniques de visualisation selon D. A. Keim en fonction des particularités des données et du contexte d'utilisation 59

LISTE DES FIGURES

2.1 Exemple d'image multispectrale CASI 11
2.2 Décomposition d'une image couleur en image à 3 composantes 14
2.3 L' image multicomposante : un ensemble de données multidimensionnelles particulières 15
2.4 Volume de l'hypersphère unité, en fonction de la dimension de l'espace . . 19
2.5 Ratio du volume de la sphère unité sur le volume du cube unité, en fonction de la dimension de l'espace 20
2.6 Ratio du volume de la sphère de rayon $0,9$ sur le volume de la sphère de rayon 1, en fonction de la dimension de l'espace 20
2.7 Pourcentage de points d'un échantillon gaussien contenus dans la sphère de rayon 1.65, en fonction de la dimension de l'espace 21
2.8 Images des Composantes Principales de l'image CASI 29
2.9 Pourcentage de variance expliquée par les composantes principales de l'échantillon CASI 31
2.10 Pourcentage cumulé de variance expliquée par les composantes principales de l'échantillon CASI 31
2.11 Composantes Indépendantes de l'image hyperspectrale CASI présentée sur la figure 2.1 35

3.1 *Matrice de Scatterplots* de la base de données IRIS 45
3.2 Courbes d'Andrew pour la base de données IRIS 46
3.3 Représentation en coordonnées parallèles d'un point en dimension 5 47
3.4 Coordonnées parallèles de la base de données IRIS 48
3.5 Visualisation par glyphes-étoiles (*Glyphs 'Star'*) de 15 données extraites de la base de données IRIS 50
3.6 Visualisation par glyphes-visages (*Glyphs 'Face'*) des même 15 données extraites de la base de données IRIS 51

3.7 Visualisation par *Glyphs 'Star' et 'Face'* de la base de données IRIS . . . 52
3.8 Exemple de découpages classiques en sous-fenêtres. L'exemple du haut concerne des données en dimension 6 tandis que celui du bas concerne les données en dimension 8. 54
3.9 Courbe de Peano-Hilbert en "U" (gauche) et en "Z" (droite). Ces courbes remplissent une grille carrée de 8 points de coté. 55
3.10 Coordonnées parallèles des données correspondant à l'image multicomposante de fluorescence X. La lecture est rendue quasi-impossible par le nombre de points élevé de l'échantillon (1560 individus) 60
3.11 Visualisation par Andrews Plot des données correspondant à l'image multicomposante de fluorescence X. La représentation est à nouveau rendue illisible à cause de nombre de points élevé de l'échantillon (1560 individus) 61
3.12 Visualisation par Matrice de Scatterplots des données correspondant à l'image multicomposante de fluorescence X 62
3.13 Etapes de la construction récursive d'une courbe de Hilbert (dite en "U"). 70
3.14 Image en "vraies" couleurs constituant l'ensemble de données à visualiser . 72
3.15 Image en "fausses" couleurs, fournie par notre méthode de visualisation . . 72
3.16 Luminance de l'image initiale de la figure 3.14 73
3.17 Luminance de l'image générée par notre technique (Figure 3.15) 73
3.18 Les six composantes d'un échantillon de 65.536 données (image 256×256 à six composantes) 74
3.19 Visualisation couleur de 65.536 données d'un espace de dimension six : image couleur 256×256 permettant de visualiser les 12 classes de données. 75
3.20 Les six composantes d'un échantillon de 65.536 données (image 256×256 à six composantes) 76
3.21 Les trois premières composantes principales de l'échantillon de 65.536 données. 76
3.22 Visualisation couleur de 65.536 données d'un espace de dimension six : image couleur 256×256 permettant de visualiser les 16 classes des données, non spatialement organisées. (image grossie) 76
3.23 Visualisation couleur de 65.536 données d'un espace de dimension six : image couleur 256×256 permettant de visualiser les 16 classes de données, spatialisées (image grossie). 77
3.24 Image à 14 composantes visualisée par une seule image couleur et les quatre phases détectées par une méthode de classification 79
3.25 Visualisation des 150 données IRIS de dimension 4 (image couleur et label des trois classes) 80
3.26 Visualisation des 1024 données extraite de la base "Forest Cover Type", de dimension 10 (image couleur et label des quatre classes) 81

3.27 Schéma représentant le processus de création de l'image couleur à partir des données initiales 82
3.28 Définition d'un parcours aléatoire à travers la base de données IRIS 85
3.29 Images extraites de la séquence obtenue lors du parcours défini à la figure 3.28. Ces deux images se succèdent en passant le point défini par la croix rouge de ladite figure 86
3.30 Reconstitution de la "vraie" classification à partir des frontières visibles sur les images de la figure 3.29. Seule une petite partie de cette frontière n'est pas vraiment visible. 87

4.1 Exemple de dendrogramme 96
4.2 Image couleur "Lena" 105
4.3 Segmentation d'une image couleur par Mean Shift 106
4.4 Exemple introductif : Echantillon de personnes étudiées ; les individus sont caractérisés par leur opinion sur un sujet donné. 109
4.5 Exemple introductif : Chaque individu définit ses préférences sur l'ensemble des autres individus 111
4.6 Exemple introductif : représentativité d'un individu dans l'échantillon . . 112
4.7 Exemple introductif : "parmi deux individus qu'il "préfère autant", il se connectera à l'échantillon par celui qui est le plus représentatif" 113
4.8 Exemple introductif : "parmi deux individus d'égale représentativité, il se connectera à l'échantillon par celui qu'il préfère" 113
4.9 Degré d'appartenance des données en fonction de leur rang : cas d'une fonction g Gaussienne avec $s = 40$ 115
4.10 Echantillon (en dimension 1) de 120 données simulées par deux distributions Gaussiennes $\mathcal{N}(5, 2), \mathcal{N}(60, 15)$ 116
4.11 Fonctions d'appartenance des 30ème et 90ème données de l'échantillon . . 117
4.12 Fonction d'appartenance à l'échantillon de 120 données 119
4.13 Poids utilisés pour représenter l'échantillon de 120 données 119
4.14 Construction des poids pour la procédure d'agrégation OWA 120
4.15 Connexion floue pour les 30ème et 90ème données de l'échantillon 122
4.16 Exemple 1 : Echantillon simulé , en dimension 2, de 400 données 124
4.17 Exemple 1 : Fonction d'appartenance à l'échantillon de 400 données 125
4.18 Exemple 1 : Graphe obtenu à partir de la défuzzyfication des ensembles de connexion de l'échantillon de 400 données 125
4.19 Exemple 2 : Echantillon, en dimension 2, composé de 3 classes de 200, 100 et 100 données 127
4.20 Exemple 2 : Fonction d'appartenance à l'échantillon 127
4.21 Exemple 2 : Estimation de la fonction de densité de probabilité de l'échantillon par la méthode de Parzen 128

4.22 Exemple 2 : Vecteur de poids utilisé pour l'opérateur d'agrégation 128
4.23 Exemple 2 : Graphe associé à l'échantillon, construit à l'aide des ensembles de connexion et dont les composantes connexes déterminent les classes de données 129
4.24 Exemple 3 : Echantillon, en dimension 2, composé de 3 classes de 500, 200 et 50 données 129
4.25 Exemple 3 : Fonction d'appartenance à l'échantillon. 130
4.26 Exemple 3 : Estimation de la fonction de densité de probabilité de l'échantillon par la méthode de Parzen 131
4.27 Exemple 3 : Vecteur de poids utilisé pour l'opérateur d'agrégation 131
4.28 Exemple 3 : Graphe associé à l'échantillon, construit à l'aide des ensembles de connexion et dont les composantes connexes déterminent les classes de données 132
4.29 Image à 14 composantes 132
4.30 Résultat de la classification des pixels de l'image multicomposante de Fluorescence X. Chacune des 3 classes de pixels est représentée par un niveau de gris arbitraire différent 133
4.31 Résultat obtenu avec la méthode *Herbin et al* 133

Chapitre 1

Introduction

Le développement des moyens d'acquisition et l'augmentation des capacités de stockage et de calculs ont fait naître un besoin en techniques nouvelles pour les images multicomposantes. Une image multicomposante est un ensemble de plusieurs plans représentant une même scène ou les mêmes objets, obtenus avec des paramètres d'acquisition différents. L'exemple le plus classique d'images multicomposantes est celui des images multispectrales satellitaires (LANDSAT, SPOT ou CASI par exemple). L'analyse de ce type d'images pose entre autres les questions suivantes : Comment observer l'ensemble des informations recueillies ? Comment découvrir les structures sous-jacentes de ces données ?

De façon plus générale, les problèmes soulevés par ces images sont liés à ceux rencontrés pour les données multidimensionnelles. Les questions que nous nous posons sur les images multicomposantes sont aussi valables pour des données quelconques. Les méthodes d'*analyse des données* comprennent des outils performants qui peuvent tendre à y répondre. Pour reprendre une phrase de G. Celeux et col., on peut dire que *"l'analyse des données cherche à extraire d'une grande masse de données multidimensionnelles les "informations utiles". Cette synthèse peut-être effectuée à l'aide de méthodes de visualisation et de méthodes de structuration"*. Notre problématique sur les images multicomposantes se formule donc comme un problème d'analyse de données plus général, celui de la *visualisation* et la *classification* de données multidimensionnelles. Ces deux notions sont connexes et constituent deux approches d'analyse exploratoire de données.

Le traitement des bases de données multidimensionnelles n'est pas une tâche triviale. En effet ce type de données a des particularités qui induisent des contraintes et des problèmes spécifiques. La dimensionnalité, ou dimension de l'espace des données, correspond au nombre de variables[1] (ou attributs) qui décrivent ces données. Cette dimensionnalité

[1] on parle aussi de "descripteurs"

soulève de nombreux points délicats souvent négligés.
Si l'on sait facilement se représenter et visualiser des données en dimension un, deux, ou trois, il est en revanche beaucoup moins trivial de le faire en dimension supérieure. Ce problème reste vrai pour les images multicomposantes : visualiser une image en niveau de gris (correspondant à des données en dimension un) ou en couleurs (correspondant dans ce cas à des données en dimension trois) est immédiat et ne pose aucune difficulté. Mais qu'en est-il des images en dimension quatre et plus ? La première idée est en général d'essayer d'observer les données selon chacune des variables prises séparément. Cela consiste par exemple à observer un à un les plans en niveaux de gris constituant l'image multicomposante. Si cette approche peut, moyennant des efforts de "reconstruction" (ou l'utilisation de la vidéo ou du son par exemple), convenir pour une dimension de quatre ou de cinq, il parait difficile et même impossible de reconstituer l'information globale lorsque la dimensionnalité augmente au delà.
L'autre problème de la dimensionnalité est de nature théorique. En effet le comportement des distances, des volumes, des distributions, est très déroutant lorsque l'on est en dimension très élevée. Cette particularité, appelée *malédiction de la dimensionnalité* pose de gros problèmes pour l'estimation, l'approximation ou l'évaluation d'objets, dans les espaces de grande dimension.
La dimensionnalité des données constitue donc un obstacle avec lequel il faut composer lorsque l'on souhaite analyser ce type de données. La solution habituelle consiste à réduire cette dimension à l'aide d'outils adaptés comme l'Analyse en Composantes Principales ou l'Analyse en Composantes Indépendantes pour ne citer que ces deux méthodes classiques.

Lorsque l'on souhaite observer et visualiser des données multidimensionnelles, on peut avoir recours à des méthodes statistiques appelées méthodes factorielles qui fournissent des représentations et des réductions de l'information contenue dans les données. Le résultat peut alors être fourni sous forme de représentations géométriques. Ces méthodes ne sont pas "purement graphiques". Ce sont des méthodes d'analyse qui relèvent de l'analyse linéaire. L'affichage sous forme graphique n'est pas leur finalité première. L'utilisation de l'outil informatique a permis de voir l'apparition de méthodes spécifiquement dédiées à la visualisation des données. Ces méthodes sont nombreuses et variées et permettent de visualiser les données sur différents supports. Nous nous intéressons plus particulièrement aux méthodes statiques, fournissant des représentations sur des périphériques en deux dimensions (écran ou papier par exemple). Ces méthodes utilisent divers modes de représentations utilisant de la couleur, des courbes, des nuages de points, des glyphes ou des métaphores. Malheureusement, la plupart de ces méthodes deviennent inefficaces lorsque la dimension de l'espace augmente ou lorsque le nombre de données est grand. Enfin, celles qui échappent à ces problèmes ne sont généralement pas du tout adaptées aux images multicomposantes. En effet, une méthode de visualisation dédiée à ce type

de données doit exploiter la spécificité des images, à savoir la spatialisation des données. Les pixels sont spatialement organisés dans une image. Cette information de localisation est cruciale et permet au cerveau humain de se représenter instantanément une scène ou des objets.

La seconde question posée en introduction concernait la structure des données. Comment rechercher les structures inhérentes aux données ?

La classification automatique [2] des données permet de fournir une vue concise et structurée des données. Le principe de la classification automatique est de chercher à regrouper les données à partir de leurs observations. Autrement dit, cela consiste à créer des classes parmi ces données. La classification automatique est utilisée dans de multiples disciplines et fait l'objet de recherches nombreuses. On parle par exemple de taxonomie en botanique, et d'apprentissage non-supervisé en Intelligence Artificielle. Comme l'expliquent Celeux et al, "l'information apportée par une classification se situe au niveau sémantique" et Lance et Williams "il ne s'agit pas d'atteindre un résultat vrai ou faux, probable ou improbable, mais seulement profitable ou non profitable". On rejoint ici la notion d'information utile évoquée au début de cette introduction. Le résultat d'une classification permet de mettre en évidence le pouvoir séparateur ou non des descripteurs et l'observation des classes offre une vue concise et structurée des données. Comme la littérature sur le sujet, les méthodes de classification sont nombreuses. Pour la classification des pixels d'une image multicomposante, la méthode idéale doit :

- rester efficace avec un grand nombre de données,
- pouvoir détecter des classes de formes quelconques (c'est à dire généralement non convexes),
- de densités et d'effectifs différents,
- pouvoir être robuste au bruit,
- composer avec le caractère ambigu de certains pixels.

Les techniques de classification permettent de prendre en compte les deux derniers points, tandis que les méthodes basées sur l'estimation de la densité de probabilité des données permettent de détecter les classes de formes non convexes. Des algorithmes adaptés sont également efficaces lorsque le nombre de données est important. Des méthodes adaptatives déterminent des classes de densités et d'effectifs différents et enfin, certaines améliorations permettent d'obtenir des techniques robustes au bruit. Le développement de méthodes réunissant le maximum de ces "qualités" est l'objet de nombreuses recherches.

Ce mémoire présente le résultats de travaux réalisés au cours d'une thèse de doctorat financée par une allocation de recherche ministérielle, qui s'est déroulée au CReSTIC

[2] En anglais, le terme *Clustering* désigne la classification tandis que *classification* concerne les problèmes d'analyse discriminante.

(Centre de Recherche en Sciences et Technologies de l'Information et de la Communication) de l'Université de Reims Champagne-Ardenne.
Ces travaux sont ciblés sur les deux problématiques citées au début de cette introduction et ont conduit à proposer une solution à chacune des questions initialement posées. Le travail de recherche effectué a ainsi pu couvrir les deux notions de visualisation et de structuration qui définissent, selon Celeux et al, l'Analyse des données.

Une méthode a tout d'abord été développée pour la visualisation des images multicomposantes. Elle utilise l'image couleur pour présenter une vision immédiate et synthétique des informations contenues dans les données. Une généralisation à la visualisation de données multidimensionnelles quelconques (c'est à dire non spatialisées) a ensuite été conçue. Le principe de notre méthode est de réduire la dimensionnalité par une analyse en composantes principales puis d'affecter une couleur à chaque donnée. Dans le cas d'une image multicomposante, les couleurs sont attribuées à chaque pixel multidimensionnel, et on obtient alors, en conservant la même organisation spatiale, une image couleur de la scène représentée. Dans le cas de données non spatialisées, on fait correspondre, à chaque donnée, un pixel couleur dans une image. Ces pixels sont arrangés, dans l'image produite, en utilisant une courbe de remplissage de Peano.

En classification automatique, la contribution de ce travail se situe surtout en amont du processus. Un nouveau concept de représentation des données a été élaboré. Cette représentation repose sur l'utilisation de la théorie des ensembles flous. Chaque donnée et l'ensemble lui-même sont considérés comme des ensembles flous. Une notion de connexion floue de chaque donnée est ensuite introduite. Cette représentation entend généraliser l'utilisation du flou en classification. En effet, l'approche classique consiste à définir les classes comme des ensembles flous. Chaque donnée appartient ainsi à toutes les classes, avec des degrés différents. L'approche proposée dans ce travail introduit cette notion de flou avant même le processus de classification proprement dit, en définissant chaque donnée ainsi que l'échantillon comme des ensembles flous.
L'élaboration de cette technique de représentation fait appel à des outils et techniques empruntés à divers domaines : statistiques non-paramétriques, théorie des ensembles flous, agrégation multicritère... Le caractère général des techniques utilisées permet facilement d'envisager de l'utiliser dans d'autres contextes que celui de la classification.
Nous avons développé un algorithme simple de classification utilisant cette représentation afin d'en montrer l'intérêt et l'efficacité dans ce cadre. Nous l'avons appliqué à des images multicomposantes et utilisé dans le cadre d'un processus exploratoire de données.

Ce mémoire comporte cinq chapitres dont celui correspondant à cette introduction. Le chapitre 2 est consacré à la présentation des données multidimensionnelles et du pro-

blème des grandes dimensionnalités. Après avoir expliqué ce que sont les données multidimensionnelles et les images multicomposantes, les problèmes induits par les grandes dimensionnalités sont exposés et illustrés. Les approches classiques pour réduire cette dimension de l'espace des données sont alors présentées avec des illustrations d'utilisation pour les images multicomposantes.
Dans le chapitre 3, une solution au problème de visualisation des données multidimensionnelles est proposée. Après avoir survolé les méthodes classiques de visualisation, nous présentons notre méthode en détails. Les processus d'affectation des couleurs et d'organisation spatiale des pixels y sont expliqués. Des applications de validation du principe sont également exposées. Puis, au delà des travaux de validation sur des données simulées, nous utilisons notre technique sur des bases de données réelles et de "vraies" images multicomposantes. Son utilisation en vue de l'exploration dynamique de masses de données est enfin proposée.
Le chapitre 4 présente une nouvelle approche pour la classification de données. Les algorithmes et familles d'algorithmes classiques sont d'abord exposés puis notre méthode de représentation est détaillée : un premier exemple introductif permet d'appréhender, de façon plus intuitive et indépendante du contexte de classification, les nouvelles notions utilisées ; ces notions sont ensuite présentées plus formellement et illustrées. Ensuite un premier algorithme de classification utilisant notre méthode de représentation est présenté et des applications à des données simulées, une base de donnée réelle et une image multicomposante proposées.
Enfin, le chapitre 5 présente des éléments de discussion, des perspectives et une conclusion à ce travail.

Chapitre 2

Données multidimensionnelles - Grande dimensionnalité

Pour écrire, il faut aimer, et pour aimer il faut comprendre.
John Fante (Mon chien stupide)

La chose importante est la chose évidente que personne ne dit.
Charles Bukowski (Le ragoût du septuagénaire)

Sommaire

2.1 Les données multidimensionnelles 9
2.1.1 Exemples de données multidimensionnelles 9
2.1.2 Cas particulier des images multicomposantes 13
2.2 Malédiction de la dimensionnalité 14
2.2.1 Problème de représentation 18
2.2.2 Phénomène d'espace vide (ou d'espace creux) 18
2.2.2.1 Volume de la sphère unité 18
2.2.2.2 Volumes de la sphère unité et du cube circonscrit . . 19
2.2.2.3 Volumes de deux sphères 19
2.2.2.4 Distributions Gaussiennes 19
2.2.3 Phénomène de Hughes 21
2.2.4 Hypothèse de normalité 22
2.3 Dimension intrinsèque 22
2.3.1 Méthodes locales 23

2.3.2 Méthodes globales 23
2.4 Méthodes de réduction de dimensionnalité 25
2.4.1 Méthodes linéaires 25
2.4.1.1 Analyse en Composantes Principales 25
2.4.1.2 Poursuite de Projection 32
2.4.1.3 Analyse en Composantes Indépendantes 32
2.4.2 Méthodes non-linéaires 34
2.4.2.1 Positionnement multidimensionnel 34
2.4.2.2 Cartes de Kohonen 34
2.4.2.3 Analyse en composantes curvilinéaires 36
2.4.2.4 Autres méthodes 36
2.5 Conclusion 37

L'augmentation des capacités de stockage et l'amélioration des moyens d'acquisition ont conduit à manipuler et analyser des ensembles de données de taille et de dimensionnalité toujours plus importantes. Ce constat est particulièrement vrai dans le domaine de l'image où la résolution et la variété des spectres d'acquisition sont de plus en plus grandes.

La dimensionnalité d'un ensemble de données correspond au nombre de descripteurs (les variables) des observations de l'ensemble. Lorsqu'elle est élevée, la dimensionnalité d'un échantillon de données pose de nombreux problèmes pour la visualisation et l'analyse de cet échantillon. Dans le cas d'une image hyperspectrale par exemple, les données sont les pixels de cette image et la dimensionnalité correspond au nombre de bandes spectrales différentes qui ont été utilisées pour l'acquisition. On dispose donc, lorsque l'on parle d'une image hyperspectrale, d'une séries d'images en niveaux de gris, générées par le capteur et représentant la même scène. Ces nombreuses images, représentant les mêmes objets avec différents paramètres d'acquisition, posent un problème pratique : comment extraire efficacement les informations utiles ? Comment se représenter globalement l'objet ?

Les problèmes liés à la dimensionnalité ne sont pas seulement de nature pratique, mais aussi théorique. En effet, s'il semble compliqué d'observer des données en dimension élevée, analyser la structure d'un tel échantillon avec des outils classiques pose également des problèmes théoriques : malédiction de la dimensionnalité, phénomène de Hughes, etc... Il est donc souvent nécessaire de réduire la dimensionnalité d'un problème lors de tout processus exploratoire ou analytique de données multidimensionnelles. Il existe d'ailleurs de nombreuses méthodes pour réduire la dimensionnalité. Ces méthodes sont présentées, dans ce mémoire, en deux catégories : les méthodes linéaires et les méthodes non linéaires.

Nous allons tout d'abord décrire de façon générale, dans la section 2.1, ce que l'on appelle *données multidimensionnelles* et présenter quelques exemples, puis, dans une seconde partie 2.2, les problèmes et les difficultés induites par la *dimensionnalité* seront exposés et illustrés à l'aide de plusieurs exemples. La partie 2.3 est consacrée à la notion de *dimensionnalité intrinsèque*. Enfin, la section 2.4 présente des méthodes de *réduction de dimensionnalité* avant qu'une conclusion de ce chapitre ne soit proposée.

2.1 Les données multidimensionnelles

Les données multidimensionnelles peuvent être vues comme des ensembles d'*individus* (ou *observations*) décrits par plusieurs *variables* (ou paramètres) : les *descripteurs*. On appelle *dimensionnalité* la dimension de l'espace des individus, donc le nombre de descripteurs.

Nous allons d'abord présenter des exemples de données multidimensionnelles, puis nous présenterons le cas particulier des images multicomposantes.

2.1.1 Exemples de données multidimensionnelles

Les avancées technologiques ont permis le développement de nouvelles techniques et matériels d'acquisition d'images, que ce soit dans le domaine médicale, en astronomie ou en biologie. De manière plus générale, l'industrie des technologies de l'information bénéficie d'une croissance rapide et constitue un segment très lucratif de l'économie. Cette croissance se traduit par la multiplication de masses de données considérables dans tous les secteurs d'activité (scientifique, médical, ingénierie, économique, etc...) On peut citer exemple :

Données biotechnologiques. L'étude des données sur le génome humain a connu, ces dernières années, de fantastiques progrès. L'analyse de ce type de données fait l'objet de nombreux et actifs travaux de recherche. (Exemple : `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/molecular-biology`)

Données médicales. Les données sur les patients et les pathologies constituent de grandes masses de données qu'il est intéressant d'explorer et analyser pour développer des système explicatifs ou prédictifs. (Un exemple de base de données sur l'arythmie cardiaque est disponible à l'adresse : `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/arrhythmia/` ou sur le cancer du sein à `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin`).

Données financières. Le nombre de données disponibles sur les transactions financières et sur l'évolution des marchés est énorme et l'analyse de telles bases nécessite des

outils adaptés pour construire des modèles et des systèmes de prédiction aussi efficaces que possible.

Imagerie satellite. Le nombre important de satellites envoyés rend disponible un grand nombre d'images ou de séquences d'images. Ce type d'imagerie est par exemple très utile pour l'étude et la découverte des ressources naturelles ou pour l'exploitation des sols. (Exemple de bases d'images : http://aviris.jpl.nasa.gov/html/aviris.freedata.html)

Imagerie hyperspectrale. Ce type d'imagerie est de plus en plus répandu, par exemple sous forme d'imagerie aérienne ou satellitaire. Les capteurs hyperspectraux permettent de capturer des images sur des bandes spectrales très variées (en sus des classiques canaux R,G,B). Ce type d'images permet par exemple de révéler les compositions chimiques de l'élément observé. (Exemple de bases d'images : http://aviris.jpl.nasa.gov/html/aviris.freedata.html)

Données de clientèle (vente, marketing). Toutes les données concernant les achats et les habitudes des clients sont enregistrées et analysées. Ce type de recueil d'information est notamment facilité par le développement important de l'achat en ligne. L'analyse de ces masses considérables d'information permet par exemple le ciblage de la clientèle et permet aux vendeurs d'adapter leurs méthodes de ventes et leurs produits.

On peut trouver des exemples de bases de données variées et bien documentées [150] à l'URL : http://www.ics.uci.edu/~mlearn/MLSummary.html

Dans ce travail de thèse, les données considérées sont quantitatives et on s'intéresse également plus précisément aux données particulières que constituent les images multidimensionnelles. Voici deux exemples de types de données multidimensionnelles (données quelconques non-spatialisées, et données images, spatialisées) que nous utiliserons dans ce travail :

base de données IRIS de Fisher [68][150] Cette base de données "classique" [12] contient la description de 150 fleurs appartenant à trois classes d'iris connues : *virginica*, *versicolor* et *setosa*. Chaque fleur est décrite par les longueurs et largeurs de ses pétales et sépales, soit 4 descripteurs qui permettent de définir à quelle classe d'iris elle appartient. Cette base de données est extrêmement classique et très souvent étudiée. Elle permet par exemple de tester les méthodes de classification puisque les classes de données sont connues. Les classes sont caractérisées par le fait que l'une d'entre elles est linéairement séparable des deux autres qui ne le sont pas entre elles [150].

images multispectrales CASI [1] Le CASI (Compact Airborne Spectrographic Ima-

[1] Cette image est présentée à titre d'exemple et représente un type classique d'image multidimensionnelle

ger) est un spectrographe permettant d'acquérir des images multispectrales dans le visible et le proche infra-rouge (gamme de longueur d'onde allant de 400 à 900 nm avec un angle de visée de 42° en travers de la ligne de vol). L'ensemble des données est alors constitué de plusieurs images de la même scène, chaque image correspondant à une bande spectrale différente. Un pixel est un individu, et les descripteurs sont les valeurs du niveau de gris correspondant à ce pixel dans chaque image acquise. Les images de la figure 2.1 sont issues du capteur hyperspectral CASI du GSTB[2] (394 – 907 nm), embarqué dans un avion, qui permet d'acquérir des images avec des résolutions très fines à partir de 0.50 cm.

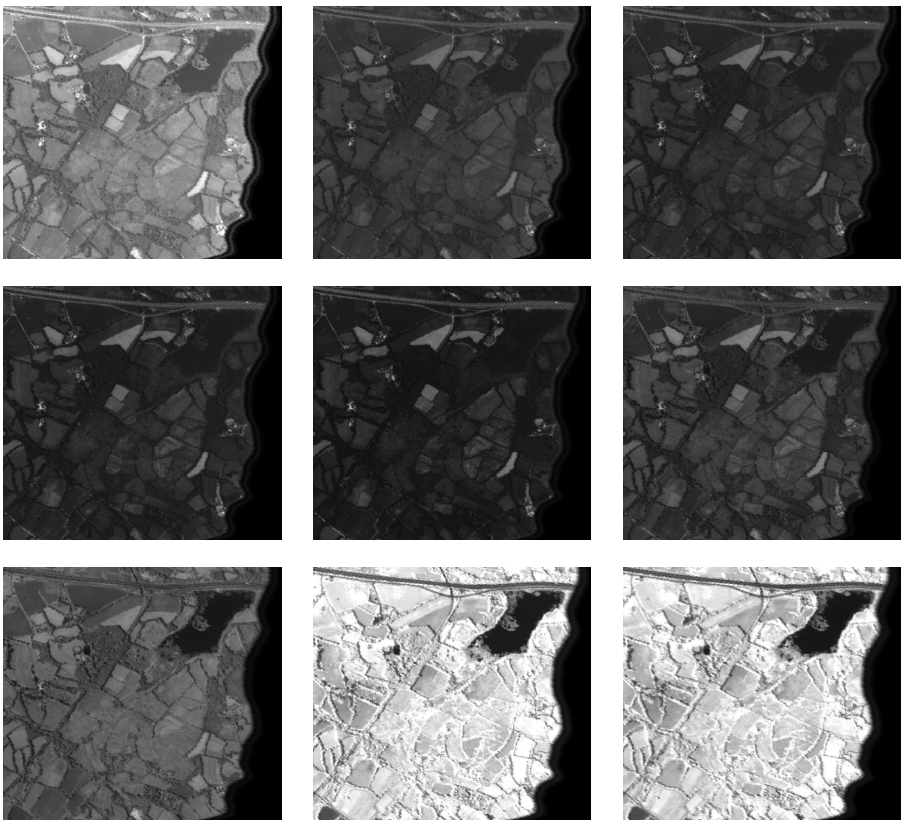


FIG. 2.1 – Exemple d'image multispectrale : 9 composantes d'une image multi-spectrale de la baie de Lannion acquises par le CASI du GSTB. La résolution retenue est de 4. Les conditions d'acquisition de chaque image sont indiquées dans la table 2.2

[2]Groupement Scientifique de Télédétection en Bretagne

Longueur sép.	Largeur sép.	Longueur pét.	Largeur pét.	Type d'iris
5,1	3,5	1,4	0,2	Iris-setosa
4,9	3,0	1,4	0,2	Iris-setosa
4,7	3,2	1,3	0,2	Iris-setosa
4,6	3,1	1,5	0,2	Iris-setosa
		(...)		
5,1	3,8	1,9	0,4	Iris-setosa
4,8	3,0	1,4	0,3	Iris-setosa
5,1	3,8	1,6	0,2	Iris-setosa
4,6	3,2	1,4	0,2	Iris-setosa
7,0	3,2	4,7	1,4	Iris-versicolor
6,4	3,2	4,5	1,5	Iris-versicolor
6,9	3,1	4,9	1,5	Iris-versicolor
5,5	2,3	4,0	1,3	Iris-versicolor
		(...)		
5,7	2,9	4,2	1,3	Iris-versicolor
6,2	2,9	4,3	1,3	Iris-versicolor
5,1	2,5	3,0	1,1	Iris-versicolor
5,7	2,8	4,1	1,3	Iris-versicolor
6,3	3,3	6,0	2,5	Iris-virginica
5,8	2,7	5,1	1,9	Iris-virginica
7,1	3,0	5,9	2,1	Iris-virginica
6,3	2,9	5,6	1,8	Iris-virginica
		(...)		
6,3	2,5	5,0	1,9	Iris-virginica
6,5	3,0	5,2	2,0	Iris-virginica
6,2	3,4	5,4	2,3	Iris-virginica
5,9	3,0	5,1	1,8	Iris-virginica

TAB. 2.1 – Base de données *Iris de Fisher*

Numéro de bande	centre de bande (nm)	Largeur de bande (nm)
1	551.1	8.4
2	571.1	10.2
3	600.9	8.4
4	636.5	8.4
5	677.5	8.4
6	696.5	17.4
7	715.4	8.4
8	749.5	8.4
9	799.9	8.4

TAB. 2.2 – Conditions d'acquisition des images de la figure 2.1.

L'augmentation du nombre de variables d'un échantillon a donc tendance à apporter plus d'information sur cet échantillon. On peut donc penser que plus la dimensionnalité d'un problème est élevée et plus nous disposerons d'information.
Malheureusement, un nombre élevé de descripteurs pose de nombreux problèmes dans les processus d'exploration et d'analyse des échantillons multidimensionnels et lorsque ce nombre est trop élevé, paradoxalement, il dégrade l'information. C'est ce que nous verrons dans la partie 2.2 consacrée aux problèmes engendrés par une dimensionnalité élevée.

2.1.2 Cas particulier des images multicomposantes

Les images multicomposantes (comme les images multispectrales par exemple [132]) sont des données multidimensionnelles particulières et ont donc des caractéristiques qui leur sont propres (voir [137]). La première caractéristique est la spatialisation des données. Les données -les pixels- sont organisés spatialement dans une image. Chaque individu de l'échantillon (i.e. chaque pixel) possède donc une position dans l'image. Cette organisation est importante et ces informations peuvent parfois être considérées comme des descripteurs supplémentaires. On peut par exemple ajouter deux descripteurs correspondant aux coordonnées des pixels dans l'image. On peut ainsi représenter les données CASI sous forme d'un tableau à 11 colonnes (voir table 2.4).
On considérera dans ce travail que les composantes d'une image multicomposante sont des images en niveau de gris.
L'image couleur peut ainsi être considérée comme une image multicomposante ayant 3 composantes. La couleur peut en effet être décomposée en trois composantes : Rouge, Vert et Bleu, comme l'illustre la figure 2.2.
Chaque pixel d'une image multicomposante est un individu de l'échantillon et chaque

composante correspond à une variable. La valeur d'une variable pour un pixel donné correspond à la valeur du niveau de gris du pixel considéré sur la composante correspondante (Figure 2.3). L'image multicomposante CASI présentée sur la figure 2.1 est représentée par le tableau de données à 638.556 lignes (correspondant aux 838×762 pixels) et 9 colonnes (correspondants aux 9 images composantes) de la Table 2.3.

Image couleur

Image du canal Rouge

Image du canal Vert

Image du canal Bleu

FIG. 2.2 – Décomposition d'une image couleur en image à 3 composantes

Après cette présentation de ce que sont les données multidimensionnelles nous allons présenter les problèmes engendrés par les dimensionnalités élevées.

2.2 La malédiction de la dimensionnalité (*curse of dimensionality*)

Le terme de "*Curse of Dimensionality*" [59] a été pour la première fois employé par Richer Bellman [6] à propos de la difficulté d'un problème d'optimisation dans un espace produit, par énumération exhaustive. Il donnait l'exemple suivant :
Si on considère un découpage d'espacement 1/10 du cube unité, en dimension 10, on obtient 10^{10} points. En dimension 20, on obtiendrait 10^{20} points. Bellman explique alors qu'optimiser une fonction sur un espace produit discrétisé, en grande dimension, nécessiterait un nombre d'évaluations de la fonction extravagant.

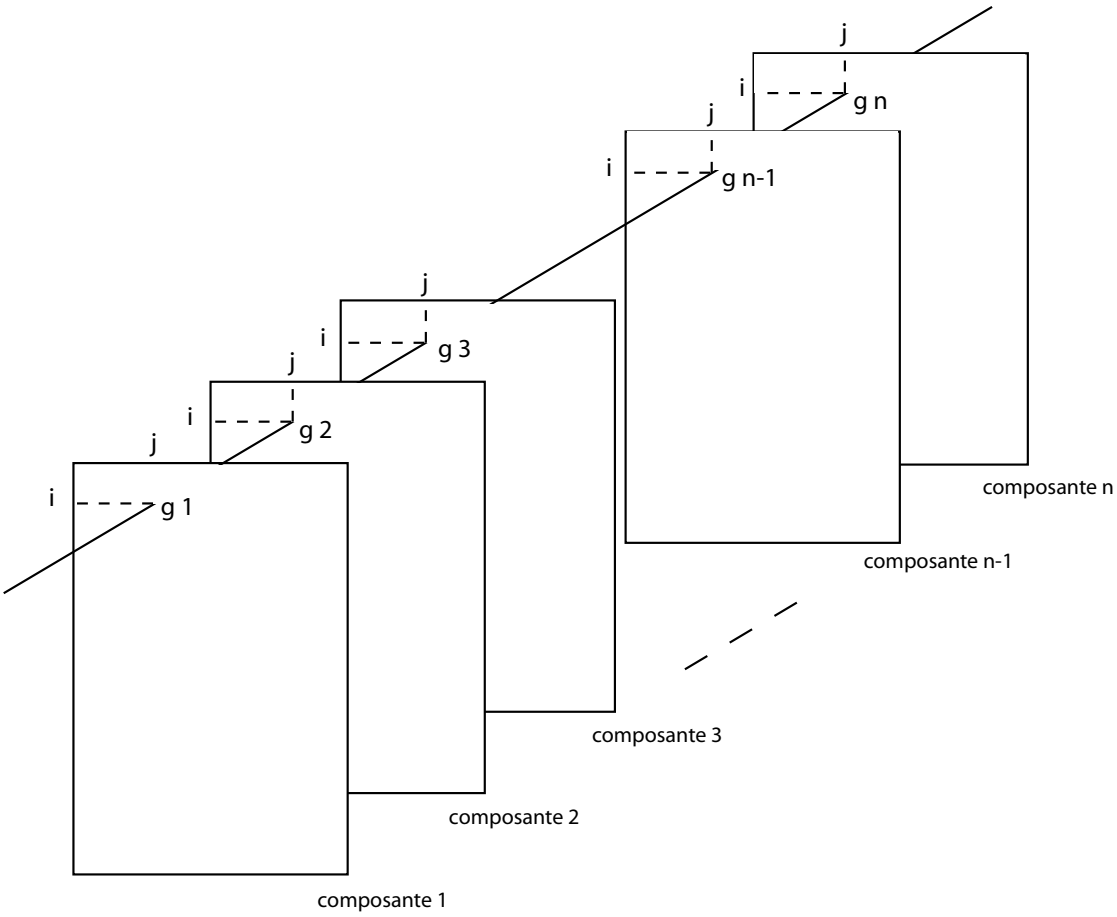


FIG. 2.3 – L' image multicomposante : un ensemble de données multidimensionnelles particulières

193	91	86	83	78	75	73	99	88
199	92	89	84	79	76	75	100	90
199	96	91	89	83	78	76	102	91
168	78	75	72	67	62	62	84	76
160	75	72	67	64	62	60	81	74
162	77	73	67	64	60	59	81	71
174	84	79	74	67	84	101	183	169
					...			
					...			
					...			
62	26	20	16	13	12	14	25	20
59	25	20	15	12	12	13	24	19
55	23	19	16	13	13	13	23	18
27	12	11	10	8	8	10	18	15
					...			
					...			
					...			

TAB. 2.3 – Extrait du tableau de données correspondant aux images CASI de la figure 2.1

193	91	86	83	78	75	73	99	88	1	1
199	92	89	84	79	76	75	100	90	1	2
199	96	91	89	83	78	76	102	91	1	3
168	78	75	72	67	62	62	84	76	1	4
160	75	72	67	64	62	60	81	74	1	5
162	77	73	67	64	60	59	81	71	1	6
174	84	79	74	67	84	101	183	169	1	7
					...					
					...					
					...					
62	26	20	16	13	12	14	25	20	185	370
59	25	20	15	12	12	13	24	19	185	371
55	23	19	16	13	13	13	23	18	185	372
27	12	11	10	8	8	10	18	15	185	373
					...					
					...					
					...					

TAB. 2.4 – Extrait du tableau de données correspondant aux images CASI de la figure 2.1, avec les deux variables de localisation (deux dernières colonnes)

Dans [59], il est montré que, dans un problème d'approximation de fonction, sans faire d'hypothèse sur la fonction de d variables, il est nécessaire de procéder à un nombre d'évaluations de l'ordre de $(1/\varepsilon)^d$ pour obtenir une erreur d'approximation ε.
La malédiction de la dimensionnalité se manifeste sous de nombreuses formes et perturbe notre intuition en changeant le comportement et la représentation que l'on peut se faire des distances, des distributions ou des volumes dans ces espaces de grande dimension (on pourra trouver d'autres exemples et études de ces problèmes dans [1][59][132][191],[53] ou encore dans [114]). Nous allons illustrer ce phénomène en évoquant certaines de ses manifestations et conséquences.

2.2.1 Problème de représentation

Le premier problème soulevé par les données en dimension supérieure à 3 est lié à la représentation mentale de ces données par le cerveau humain. En effet, l'oeil et le cerveau nous permettent de représenter facilement des données en dimension 2 ou 3 (éventuellement 4 si l'on fait appel à des animations), cette représentation devient compliquée lorsque la dimension est plus élevée. Elle peut même devenir impossible et nécessite de faire appel à des avatars, glyphes ou métaphores qui tentent de nous aider à représenter les données dans des espaces de dimension supérieure à trois (voir section 3.1.2.4). Par ailleurs, si la tentative de généralisation de la représentation $2D$ ou $3D$ à une dimension plus élevée est parfois impossible, elle est également dangereuse conduisant l'utilisateur à se faire des idées erronées, comme nous allons voir par la suite.

2.2.2 Phénomène d'espace vide (ou d'espace creux)

Lorsque la dimensionnalité d'un problème est élevée (i.e., pour rappel, lorsque la dimension de l'espace des individus dépasse 3), le comportement et les représentations des objets, peuvent sembler surprenants. Les résultats qui suivent montrent que la généralisation du cas $3D$ au cas nD n'est pas triviale. Quelques exemples vont nous permettre d'expliquer ce fait. Nous avons repris dans ce mémoire des résultats classiques que l'on peut aussi trouver dans [191] et [113].

2.2.2.1 Volume de la sphère unité

Considérons tout d'abord le volume de la sphère unité en dimension d. On a :

$$V_d = \frac{\pi^{d/2}}{\Gamma(d/2+1)}.r^d \tag{2.1}$$

Si on représente graphiquement l'évolution de ce volume en fonction de la dimension de l'espace, on constate (Figure 2.4) que le volume de la sphère unité (hypersphère unité)

décroît vers 0 lorsque d augmente (à partir de $d = 5$). Ce constat est assez difficile à imaginer si on tente de l'appréhender par une représentation mentale graphique intuitive.

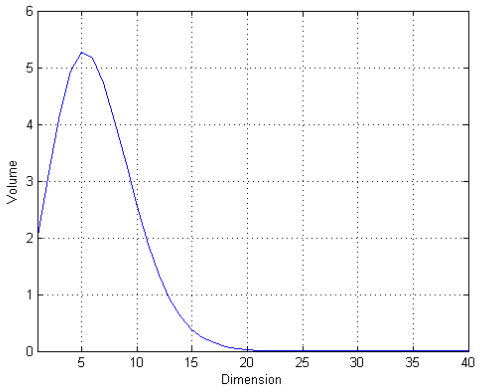


FIG. 2.4 – Volume de l'hypersphère unité, en fonction de la dimension de l'espace

2.2.2.2 Volumes de la sphère unité et du cube circonscrit

Ce comportement surprenant se poursuit lorsque l'on calcule le ratio entre le volume de la sphère unité et le volume du cube d'arrête égale au diamètre de la sphère unité (autrement dit le cube "circonscrit" à cette sphère). On remarque en effet (Figure 2.5) que, certes ce ratio décroît, mais qu'il est inférieur à 10% en dimension supérieure 6. Autrement dit, le volume se concentre "dans les coins" lorsque la dimension est élevée.

2.2.2.3 Volumes de deux sphères

L'étude [191] du ratio entre le volume de la sphère de rayon 0.9 et la sphère de rayon 1 nous montre que 90% du volume de la sphère, en dimension supérieure à 20, est contenu dans la partie dont l'épaisseur ne représente que 10% du rayon.

2.2.2.4 Distributions Gaussiennes

Enfin, ce dernier exemple permet d'illustrer le comportement d'une distribution Gaussienne en grande dimension. Cet exemple est d'autant plus intéressant que les noyaux gaussiens sont très utilisés dans les méthodes d'approximation ou d'estimation de fonctions. En dimension 1, 90% des points d'un échantillon suivant une loi normale sont contenus dans l'intervalle $[-1,65; 1,65]$. Lorsqu'on augmente la dimension, le pourcentage de points contenus dans la sphère de rayon $1,65$ décroit (Figure 2.7). En dimension

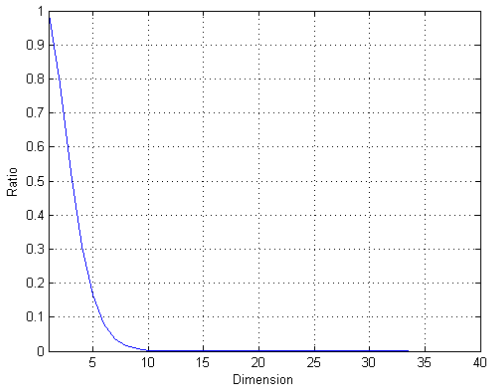


FIG. 2.5 – Ratio du volume de la sphère unité sur le volume du cube unité, en fonction de la dimension de l'espace

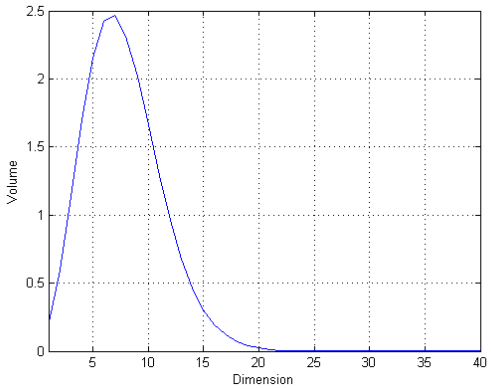


FIG. 2.6 – Ratio du volume de la sphère de rayon $0,9$ sur le volume de la sphère de rayon 1, en fonction de la dimension de l'espace

10 cette sphère contient moins de 1% des points. En grande dimension on peut donc dire autrement que les points d'un échantillon gaussien ont tendance à être dans les queues de distribution. Tous les tests statistiques classiques établis pour la dimension 1 sont remis en cause et peuvent même devenir aberrants quand la dimension est très élevée.

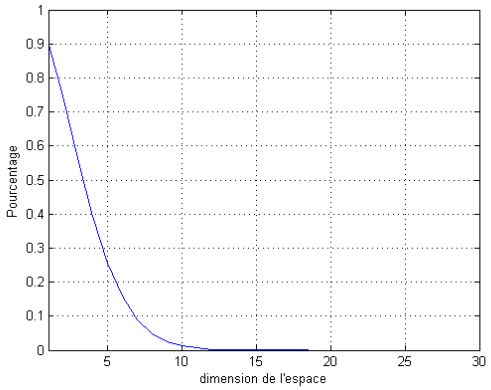


FIG. 2.7 – Pourcentage de points d'un échantillon gaussien contenus dans la sphère de rayon 1.65, en fonction de la dimension de l'espace

2.2.3 Phénomène de Hughes

Le phénomène de Hughes [101] est une autre conséquence de la *malédiction de la dimensionnalité*. C'est un cas particulier de phénomène d'*espace creux*. Ce phénomène désigne la difficulté rencontrée lors de l'estimation de fonctions, ou des paramètres d'un modèle lorsque la dimensionnalité est élevée. En effet, en grande dimension, la taille des échantillons disponibles est rarement suffisante pour pouvoir réaliser de bonnes estimations statistiques. Dans [174], Silverman pose le problème de la taille minimale d'un échantillon nécessaire dans un problème d'approximation d'une distribution Gaussienne avec des noyaux Gaussiens fixes. Les résultats de Silverman peuvent être approximés par [48] :

$$\log_{10} N(d) \approx 0,6(d - 0,25) \tag{2.2}$$

où d représente la dimension de l'espace et N la taille de l'échantillon requise.
Le nombre d'observations nécessaires dans un échantillon augmente donc exponentiellement avec la dimensionnalité. Or, dans la plupart des cas, on ne disposera pas de ce nombre minimum de points pour l'apprentissage. Cet autre problème, induit par la dimensionnalité élevée, est donc une difficulté majeure dans ce type de contexte.

2.2.4 Hypothèse de normalité

Une dernière caractéristique des espaces de grande dimension est que les projections linéaires dans des sous-espaces ont tendance à rendre les distributions gaussiennes. Si on considère un ensemble de données suivant une distribution quelconque, dans un espace de dimension p, sa projection dans un sous-espace de dimension q fixée, tend vers un modèle gaussien quand p tend vers l'infini [86].

Comme nous venons de le voir dans ces quelques lignes, la grande dimensionnalité d'un ensemble de données apporte, certes, une quantité extrêmement importante d'information, mais pose aussi quelques sérieux problèmes d'ordres pratiques et théoriques [3].
Il semble donc nécessaire, pour pouvoir appliquer aux données des méthodes classiques d'analyse et de visualisation, de devoir réduire la dimensionnalité d'un problème. En effet, la plupart des techniques d'analyse de données modernes se heurtent aux problèmes cités dans la section 2.2 lorsqu'elles nécessitent des estimations, des approximations, de l'apprentissage, etc...
Réduire la dimensionnalité d'un problème consiste donc à réduire la dimension de l'espace des individus "manipulés". Autrement dit, il s'agit de trouver un espace de dimension inférieure, dans lequel il serait plus rapide et plus efficace de manipuler les représentations données.
Avant de procéder à cette transformation des données, il apparaît légitime de se demander si la quantité d'information apportée par un nombre très important de descripteurs dans un ensemble de données, est *totalement* utile, ou s'il n'existerait pas des redondances dans l'information apportée par chacune des variables. Ces questions nous amènent donc à étudier la notion de *dimensionnalité intrinsèque* des données (ou *dimension intrinsèque*).

2.3 Dimension intrinsèque

Dans la plupart des cas de données réelles en grande dimension, ces données sont en réalité contenues dans un sous-espace de l'espace de départ, les variables pouvant être "redondantes" [190]. Fukunaga, dans [71] définit la *dimensionnalité intrinsèque* d'un ensemble de données $\Omega \subset \mathbb{R}^d$, comme la dimension M ($M < d$) du sous-espace de $\mathbb{R}^d$ contenant "en entier" tous les éléments de Ω. Estimer cette quantité est un problème complexe mais peut être intéressant. Outre les raisons citées dans les paragraphes précédents, l'estimation de cette valeur présente d'autres intérêts comme par exemple fournir

[3] à titre anecdotique, Donoho, dans [59], expose également ce qu'il appelle des *bienfaits* de la dimensionnalité

le nombre de neurones cachés lors de l'utilisations de réseaux de neurones auto-associatifs dans l'extraction non-linéaire de structures [34]. Plus généralement, la connaissance de la dimension intrinsèque permet de paramétrer les algorithmes lorsque l'on souhaite effectuer une réduction de dimensionnalité. Malheureusement, les contraintes pratiques des problèmes d'analyse d'image ou de reconnaissance de forme imposent généralement cette dimension et calculer la dimension intrinsèque n'est dans ce cas pas très utile (sauf si la méthode de réduction le nécessite).
Il existe de nombreuses méthodes d'estimation de la dimension intrinsèque de données multidimensionnelles [111][33]. On peut classer ces méthodes en deux catégories [111]. Ces deux catégories se caractérisent par des approches différentes. Dans la première catégorie la dimension intrinsèque est estimée en utilisant l'information contenue dans les voisinages de chaque point sans projeter les données dans un espace de dimension inférieure. La seconde approche est globale et ces méthodes utilisent donc l'ensemble dans sa globalité. Pour ces raisons, nous parlerons donc de méthodes locales et globales.

Nous allons présenter les concepts de ces deux types de méthodes en citant quelques algorithmes et méthodes permettant de les calculer. Nous discuterons enfin l'intérêt de cette notion dans le cadre de notre travail et conclurons.

2.3.1 Méthodes locales

Les méthodes locales (ou topologiques) tentent d'estimer la dimension topologique des données [28][29]. L'algorithme le plus connu utilisant cette approche a été proposé par Fukunaga et al dans [74]. Cet algorithme consiste à diviser les données à l'aide d'un diagramme de Voronoi, puis de déterminer les valeurs propres de la matrice de covariance locale, sur chaque voisinage ainsi créé. Les valeurs propres sont normalisées (par rapport à la plus grande) et le nombre de valeurs propres normalisées supérieures à un certain seuil T déterminent la dimension intrinsèque. D'autres techniques sont basées sur les *plus proches voisins* [186][157][192][26], ou sur les TRN (Topology Representing Networks) [143][70] pour estimer la dimension "topologique".

2.3.2 Méthodes globales

La plupart des méthodes dites "globales" sont des méthodes de projection. Le principe général des techniques de projection est de trouver un sous-espace de l'espace initial, dans lequel les données sont projetées et qui minimise l'erreur de reconstruction (autrement dit, qui minimise la perte d'information après projection). Comme nous le verrons ensuite, les méthodes de projections peuvent être divisées en deux catégories : les méthodes linéaires et les méthodes non linéaires [41]. Dans la première catégorie, on peut citer l'Analyse en Composantes Principales (ACP)[115] qui fournit une estimation de la

dimension intrinsèque en examinant l'éboulis des valeurs propres mais cette technique n'est pas très adaptée puisqu'elle a tendance à surestimer la dimension intrinsèque [11]. De même, bien qu'elle soit parfois meilleure que l'ACP pour projeter les données, l'Analyse en Composantes Principales non linéaire (non linear PCA) [125][69][130] pose aussi des problèmes pour estimer la dimension intrinsèque [141] et se révèle donc inadéquate. Il existe enfin une famille de méthodes globales appelées méthodes fractales. Ces méthodes sont issues de méthodes particulièrement efficaces dans l'estimation de la dimension intrinsèque de séries temporelles [116][136][181]. Elles fournissent des estimations de la dimension intrinsèque non-nécessairement entières, comme le sont parfois les dimensions caractérisant les fractales [142]. Plusieurs définitions de la dimension fractale ont été proposées [65]. La *box-counting dimension*[152] (dont il existe plusieurs implémentations algorithmiques efficaces [80][182][124]) et la *corrélation dimension*[81] sont les plus connues.

Nous venons de voir qu'il était possible de calculer la dimension intrinsèque d'un ensemble de données. Cette étape intervient en général au début d'un processus d'analyse de données. Toutefois l'intérêt de ce calcul est à nuancer. La première raison est liée à la définition même de la dimension intrinsèque. Elle varie en fonction du domaine dans lequel on la rencontre et, comme nous venons de le constater, il existe des définitions différentes en fonction de la méthode qui permet de la calculer ce qui rend cette notion un peu confuse. La définition plus souvent rencontrée en analyse de données (quand ce ne sont pas des séries temporelles [196]) est celle qui définit cette dimension comme "le nombre de variables utiles". La notion d'utilité est subjective et dépend du traitement et de l'analyse qui vont être effectués sur ces données. La deuxième nuance est d'ordre pratique. Lorsqu'il est nécessaire de réduire la dimension d'un espace, la dimension de l'espace à obtenir est souvent une donnée du problème. Dans la suite de ce travail notamment (chapitre 3, section 3.3), nous devons réduire la dimension à 3. La connaissance de la dimension intrinsèque ne nous est donc d'aucune utilité. Enfin, la plupart des méthodes que nous avons citées précédemment sont des méthodes assez lourdes compte tenu de l'intérêt que présentent leurs résultats dans notre cadre de travail. Dans les problèmes qui nous concernent, on ne procédera donc à l'estimation de la dimension intrinsèque que dans des cas bien précis, lorsque la méthode de réduction de dimensionnalité le nécessite par exemple, ou lorsque la dimension de l'espace après réduction n'est pas fixée.

Dans la plupart des cas, lorsque la dimensionnalité est élevée, les problèmes potentiels engendrés imposent de réduire cette dimension de l'espace sans pouvoir estimer la dimension intrinsèque. Nous allons maintenant présenter des méthodes qui permettent

de le faire.

2.4 Méthodes de réduction de dimensionnalité

Lorsqu'il est nécessaire de réduire la dimensionnalité d'un ensemble de données, le choix de la méthode peut sembler vaste mais doit être fait en fonction de la nature des données et doit être le résultat d'un compromis entre la qualité de la réduction et le coût algorithmique de son utilisation.
Il existe de nombreuses méthodes de réduction de dimensionnalité [38][53][27] utilisées en statistiques, en traitement du signal ou en IA. On peut classer ces techniques en deux catégories : les méthodes linéaires et les méthodes non-linéaires. Nous allons examiner quelques unes de ces techniques, parmi les plus classiques et les plus efficaces. Pour chacune d'elles, nous en exposerons le principe, l'intérêt et le contexte d'utilisation.

2.4.1 Méthodes linéaires

Cette catégorie regroupe les méthodes dont le principe est de déterminer p variables qui soient des combinaisons linéaires des d variables initiales (avec $p < d$) :

$$s_i = \omega_{i,1}x^1 + \ldots + \omega_{i,p}x^p \qquad i = 1, \ldots, d$$

que l'on peut écrire matriciellement :

$$s = W.x$$

où W est la matrice de taille $p \times d$ des poids.
On peut récrire cette expression de la façon suivante :

$$x = A.s$$

où A est une matrice de taille $d \times p$. Les nouvelles variables s sont appelées les variables cachées.

Les plus classiques -mais aussi, souvent, les plus adaptées- sont l'Analyse en Composantes Principales (ACP) et l'Analyse en Composantes Indépendantes (ACI) et les méthodes dérivées.

2.4.1.1 Analyse en Composantes Principales

L'*analyse en composantes principales* (ACP) est une méthode d'analyse factorielle très souvent employée comme méthode exploratoire [189] ou descriptive de données.

Nous la présenterons ici dans une autre de ses finalités, la réduction de dimensionnalité [154][128], puisque c'est le but de ce chapitre. Le but de l'ACP est de représenter un tableau de données quantitatives. Elle permet d'analyser tout tableau de données statistiques à n lignes et p colonnes représentant les p observations de n individus. C'est notamment ce champ d'applications possibles très large qui fait de l'ACP une méthode très utilisée.

Comme il est expliqué dans [39], le but de l'ACP est d'obtenir une représentation du nuage de points (associé aux individus représentés dans le tableau sus-cité) dans un espace de dimension réduite de telle manière que l'inertie portée par cet espace soit la plus grande possible. L'ACP détermine pour cela les axes principaux d'inertie du nuage, autour de son centre de gravité.

Rappelons tout d'abord la définition de l'inertie. Si on note $N(I) \subset \mathbb{R}^p$ ($N(I) = \{x_1, x_2, \ldots, x_n\}$) un nuage de points de $\mathbb{R}^p$ muni de la métrique M, et si on note $p_1, p_2, \ldots, p_n$ les poids associés aux individus, alors l'*inertie* de $N(I)$ par rapport à un point a de $\mathbb{R}^p$ est définie par :

$$I_a = \sum_{i=1}^{n} p_i . d_M^2(x_i, a) = \sum_{i=1}^{n} p_i . {}^t(x_i - a) M (x_i - a) \tag{2.3}$$

où d_M est la distance associée à la métrique M. Par ailleurs, l'espace des variables $\mathbb{R}^n$ a été muni de la métrique euclidienne dite métrique des poids, et notée D_p. Lorsque les variables sont centrées, la distance entre deux variables est égale à la covariance entre ces variables. La norme associée, au carré, d'une variable est égale à la variance de cette variable.
On appelle *inertie totale du nuage* l'inertie calculée par rapport à son centre de gravité (d'après le théorème de Huygens, cette inertie est minimale). Pour des raisons pratiques, l'ACP commence toujours par un centrage des données.

Le principe de l'ACP est donc de déterminer les axes qui prennent le mieux en compte la dispersion du nuage au sens de la distance d_M. Ces axes principaux d'inertie sont appelés *axes factoriels*. Les formes linéaires associées aux opérateurs de projections sur ces axes sont appelées *facteurs principaux*. Les projections des données initiales sur les axes factoriels sont appelées *composantes principales*.

Le coeur de l'ACP consiste donc à rechercher les axes factoriels. Il est prouvé [135][167][39] que les facteurs principaux sont déduits des vecteurs propres associés aux valeurs propres de la matrice de variance-covariance des variables du tableau de données initiales. La première composante principale -celle qui porte le plus d'inertie- est associée à la plus grande

valeur propre de cette matrice. La seconde composante principale est associée à la seconde plus grande valeur propre, et ainsi de suite.

Pourcentage d'inertie expliquée Le pourcentage d'inertie expliquée permet d'apprécier la perte d'information causée par la projection. On mesure la qualité du sous-espace F_q (i.e. engendré par les q vecteurs propres associés aux q plus grandes valeurs propres) par :

$$Q = \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

où les λ_i sont les valeurs propres de la matrice de variance-covariance des variables du tableau de données.

Remarques Comme nous l'avons vu, l'ACP impose la définition préliminaire d'une métrique euclidienne M sur $\mathbb{R}^p$. Ce choix est important et les métriques les plus utilisées sont [39] :

- la métrique associée à la matrice I_p (matrice identité de taille $p \times p$) qui est utilisée lorsque les variables sont mesurées dans des unités identiques ;
- la métrique associée à la matrice D_{1/σ^2} (matrice diagonales dont les éléments sont les inverses des variances des variables correspondantes : $D_{1/\sigma^2}(i,i) = \frac{1}{\sigma_i^2}$ où σ_i^2 est la variance de la i-ème variable)) utilisée lorsque les variables ont des variances très différentes.

Lorsque l'on utilise la métrique D_{1/σ^2}, l'ACP est appelée *Transformation de Karhunen-Loeve* (surtout en imagerie). L'emploi de cette métrique revient à analyser le tableau des données centrées-réduites.

L'ACP, utilisée comme méthode de projection, consiste donc à trouver un sous-espace de l'espace des individus qui représente du mieux possible les distances entre les individus. Lorsque l'on souhaite réduire la dimensionnalité à q, on projettera les données dans l'espace de dimension q associé aux q plus grandes valeurs propres. On peut alors mesurer la qualité de la réduction en calculant les pourcentages d'inertie expliquée (c.f. exemple de la partie suivante).
D'un point de vue algorithmique, la difficulté de l'ACP est donc le calcul des éléments propres de la matrice de variance-covariance. Il existe de nombreux algorithmes pour les calculer. L'algorithme de Jacobi, par exemple, convient puisque la matrice de variance covariance est symétrique, et calcule simultanément toutes les valeurs propres de la matrice. Pourtant cette méthode n'est pas la plus adéquate et est encore moins adaptée lorsque l'on souhaite utiliser l'ACP pour réduire la dimensionnalité. En effet pour réduire la dimensionnalité d'un problème de p à q, seules les q plus grandes valeurs propres

sont nécessaires. Il est alors plus judicieux d'utiliser des algorithmes itératifs comme la méthode de la *puissance inverse*[134] (c'est ce choix que nous avons fait dans notre implémentation de l'ACP). Cet algorithme calcule les éléments propres successivement par ordre de valeurs propres décroissantes. Si l'on souhaite réduire la dimension de 128 à 3 par exemple, il n'est nécessaire que de calculer les trois premiers éléments propres, et pas les 128.

L'ACP est une méthode très classique et très utilisée, notamment pour ses applications à la visualisation. Contrairement à de nombreuses méthodes neuronales ou auto-adaptatives, l'ACP ne repose pas sur des paramètres qui influencent la convergence ou les performances de la méthode.
Le premier point important que nous souhaitons souligner, est qu'il est possible de mesurer la perte d'information de manière objective. En effet comme nous l'avons vu on peut mesurer la qualité de la réduction à l'aide des valeurs propres. Chaque valeur propre déterminant la qualité de la composante qui lui est associée.
Le second point intéressant est également lié au précédent et réside dans l'ordre d'importance des composantes. En effet l'importance des composantes principales est hiérarchisée en fonction des valeurs propres, et lorsque l'on utilise un algorithme itératif comme celui de la puissance inverse pour leur calcul, on obtient les composantes successivement, par ordre décroissant d'inertie expliquée. Cette ordre immédiat sur les composantes est intéressant notamment dans le cas de traitement d'image, comme le montre, visuellement, l'exemple qui va suivre.
Le troisième point est d'ordre pratique. Les algorithmes implémentant l'ACP sont assez rapides et plutôt légers. Comme nous l'avons vu le seul élément difficile est le calcul des éléments propres, or les algorithmes cités sont efficaces et rapides (la convergence de la méthode de la puissance inverse, notamment, est très rapide)[134].
Enfin, de part son principe assez simple, elle permet d'interpréter les résultats fournis et les processus utilisés de façon efficace.

L'inconvénient majeur de l'ACP est la linéarité de la méthode. Elle se révélera donc inappropriée pour révéler des relations non linéaires entre les variables.

Exemple d'application Les images CASI (Figure 2.1) composent une image multi-composante de dimension 9. En effectuant une ACP sur cette image, c'est à dire sur le tableau de données correspondant (voir table 2.4), on obtient les résultats présentés sur la figure 2.8.

Les variances portées par chacune des composantes sont indiquées dans la table 2.5.

On remarque que la première composante principale porte 87% de la variance. Autrement dit, cette composante résume à elle seule 87% de l'information de la série des

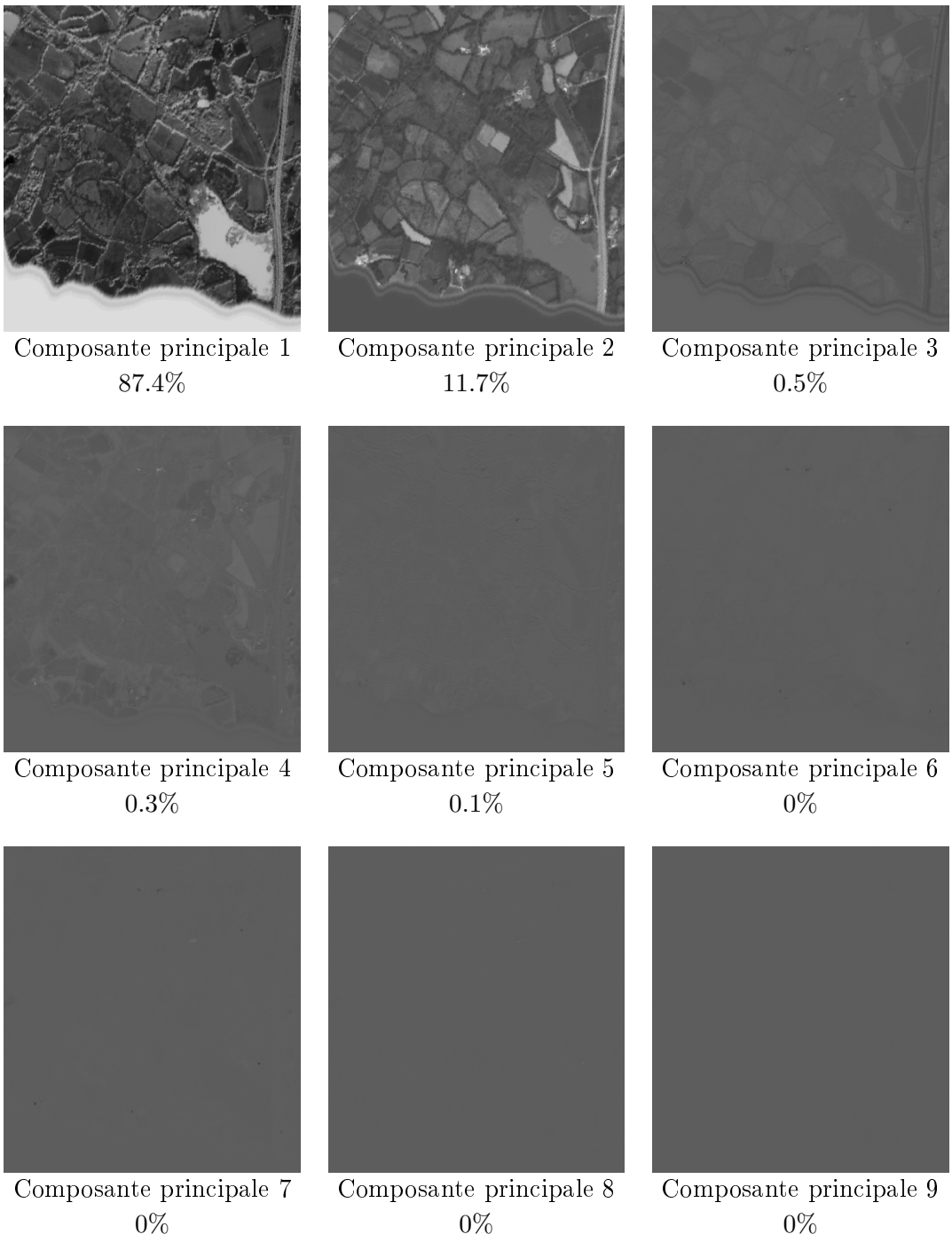


FIG. 2.8 – Images des Composantes Principales de l'image CASI, ordonnées par ordre décroissant de variance expliquée. On ne distingue plus aucune information dès la 5ème composante. Les pourcentages (arrondis à 10^{-1}) d'inertie expliquée par les composantes sont indiqués sous les images correspondantes

Composante principale	Pourcentage de variance expliquée
1	87.409
2	11.673
3	0.52205
4	0.29602
5	0.058209
6	0.022772
7	0.013354
8	0.003.8654
9	0.002.5969

TAB. 2.5 – Pourcentage de variance portée par chacune des composantes principales

Composantes principales	Pourcentage de variance expliquée cumulé
1	87.409
2	99.081
3	99.603
4	99.899
5	99.957
6	99.98
7	99.994
8	99.997
9	1

TAB. 2.6 – Pourcentage cumulé de variance portée par chacune des composantes principales

9 images initiales. On observe facilement sur la figure 2.9, qu'à partir de la composante 4, les composantes principales portent chacune une quantité d'information négligeable. Ceci se confirme lorsqu'on observe la courbe de la variance cumulée (Table 2.6), sur la figure 2.10. On voit par exemple que les trois premières composantes principales suffisent à "contenir" 99.6% de l'information apportée par les 9 composantes initiales.

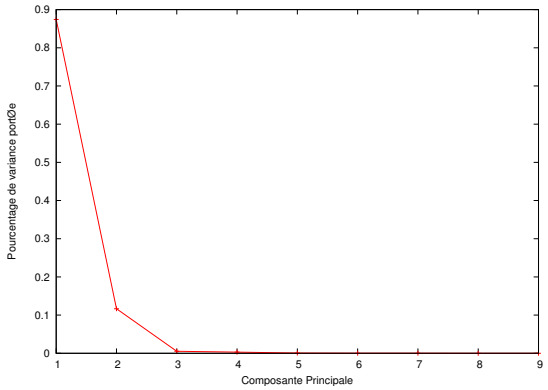


FIG. 2.9 – Pourcentage de variance expliquée par les composantes principales de l'échantillon CASI

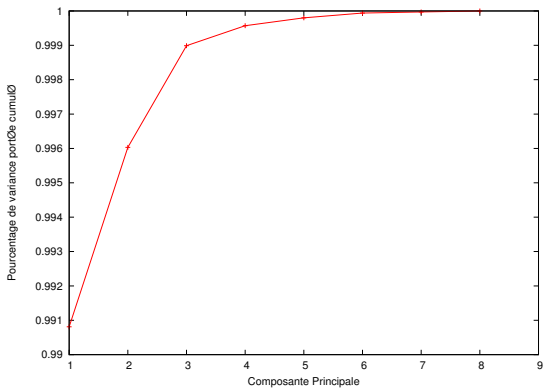


FIG. 2.10 – Pourcentage cumulé de variance expliquée par les composantes principales de l'échantillon CASI

Remarque : Dans la plupart des cas, pour les images multicomposantes, les trois premières composantes "suffisent". Ceci est vraisemblablement dû à la nature même de ces données et à la redondance induite par la structure de la plupart des images multicomposantes.

L'ACP n'est pas la seule méthode linéaire de projection. Elle peut être parfois vue comme un cas particulier de la poursuite de projection.

2.4.1.2 Poursuite de Projection

La *projection de poursuite* (ou PP) [149] est une méthode linéaire, qui contrairement à l'ACP permet d'incorporer l'information d'ordre supérieur à deux, et ainsi, est particulièrement utile pour les ensembles de données à distributions non-gaussiennes [114][160]. La PP repose tout d'abord sur la définition d'un indice de projection qui définit l'intérêt d'une direction. Le principe de la PP est donc de rechercher la direction qui maximise cet indice. En choisissant la variance comme indice, la PP rejoint alors le problème de l'ACP. L'indice le plus utilisé est basé sur l'entropie négative de Shannon. L'entropie négative d'une variable aléatoire x, de distribution de probabilité p est définie par :

$$H(x) = \int p(x) \log p(x) \tag{2.4}$$

D'autres indices de projection sont parfois utilisés, basés sur l'information de Fisher, sur les cumulants d'ordre supérieur, ou sur des mesures de non-normalité.

L'Analyse en Composantes Indépendantes est une autre approche dont un des algorithmes classiques, FastICA [106][102], permet également de calculer les directions de la poursuite de projection.

2.4.1.3 Analyse en Composantes Indépendantes

L'Analyse en Composantes Indépendantes est une méthode traditionnellement utilisée en séparation de sources. Elle permet de séparer les sources composant un mélange [47][48][103][104][105].
L'ACI est une méthode d'ordre supérieur qui consiste à chercher des projections linéaires dont les composantes, non nécessairement orthogonales (contrairement à l'ACP), sont "aussi statistiquement indépendantes" que possible. Cette condition est beaucoup plus forte que celle imposée par l'ACP (décorrélation, ou indépendance du second ordre). L'indépendance implique la décorrélation mais la réciproque n'est généralement pas vraie (sauf par exemple dans le cas de distributions Gaussiennes). L'ACI peut être vue comme une généralisation de l'ACP [47][37].

On considère le modèle (sans bruit) suivant :

$$y = Fz \tag{2.5}$$

z est un vecteur aléatoire dont les composantes maximisent une fonction de contraste (cette fonction de contraste est maximale lorsque les composantes sont statistiquement

indépendantes). Le problème résolu par l'ACI est d'estimer F et z.
L'estimation de ce modèle se réalise en deux étapes. La première consiste à définir la fonction objectif (fonction de contraste) et la seconde à optimiser cette fonction objectif. On peut classer les fonctions objectifs en deux groupes : celles qui estiment les composantes simultanément et celles qui les estiment une par une.

Dans la première catégorie, on peut citer la vraisemblance, qui conduit à estimer les paramètres du modèle en utilisant le maximum de vraisemblance. Cette méthode présente l'inconvénient d'être très sensible aux valeurs extrêmes et d'être coûteuse en calculs ce qui en fait une technique peu pratique. L'information mutuelle et la divergence de Kullback-Leiber sont plus efficaces. L'information mutuelle I mesure la dépendance de n variables aléatoires $(y_1, \ldots, y_n) = y$:

$$I(y_1, y_2, \ldots, y_n) = \sum_{i=1}^{m} H(y_i) - H(y) \tag{2.6}$$

où $H(y) = -Q(y)$, où Q représente l'entropie négative de Shannon (2.4).
L'information mutuelle est équivalente à divergence de Kullback-Leibler entre la probabilité jointe $f(y)$ et la factorisation $\hat{f}(y) = f_1(y_1)...f_n(y_n)$. On rappelle que la divergence de Kullback-Leibler est définie par :

$$\delta(f_1, f_2) = \int f_1(y) log \frac{f_1(y)}{f_2(y)} dy \tag{2.7}$$

L'estimation de l'information est malheureusement assez difficile et nécessite souvent d'utiliser des approximations [103].
On peut également évoquer les cumulants d'ordres supérieurs ou les corrélations croisées non linéaires que propose J.-F Cardoso [36]).

Dans la deuxième catégorie, la néguentropie est assez difficile à estimer :

$$J(y) = H(y_{gauss}) - H(y) \tag{2.8}$$

(où y_{gauss} est un vecteur aléatoire Gaussien ayant la même matrice de covariance que y) et les cumulants d'ordre supérieurs conduisent à des procédés qui peuvent être sensibles aux valeurs extrêmes. Le cumulant d'ordre 4 est appelé kurtosis et est défini par :

$$kurt(x) = E(x^4) - 3\left(E(x^3)\right)^2 \tag{2.9}$$

L'algorithme FastICA permet d'effectuer l'ACI (ainsi que la poursuite de projection), mais il existe de nombreux autres algorithmes (voir par exemple [117]). C'est cet algorithme que nous avons utilisé pour illustrer l'utilisation de l'ACI.

Remarque : Composantes principales et composantes indépendantes : dans le cas Gaussien, l'ACI n'apporte rien de plus que l'ACP puisqu'alors, l'indépendance se réduit à la décorrélation.

Exemple d'application L'utilisation de l'ACI pour réduire la dimensionnalité de l'échantillon des images CASI (Figure 2.1) permet de construire les images des composantes indépendantes de cet échantillon de données (Figure 2.11). Nous avons utilisé l'algorithme ICA pour générer ces images et n'avons conservé que les cinq premières composantes. L'observation de ces images confirme ce que montrait l'ACP : 3 composantes suffisent à représenter l'ensemble des 9 composantes avec une perte d'information négligeable.
On peut cependant noter que contrairement à celles obtenues avec l'ACP (figure 2.8), on n'obtient pas les composantes de façon ordonnée, ce qui ne facilite pas la lecture.

2.4.2 Méthodes non-linéaires

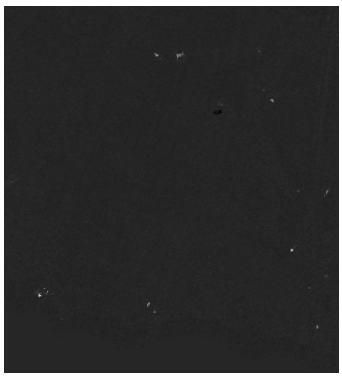
Lorsqu'il existe de fortes relations non linéaires entre les données, les méthodes linéaires classiques se révèlent assez inefficaces pour réduire la dimensionnalité. La nécessité d'utiliser des outils de réduction non linéaires [144] nous amène donc à présenter les méthodes suivantes (voir également [4]).

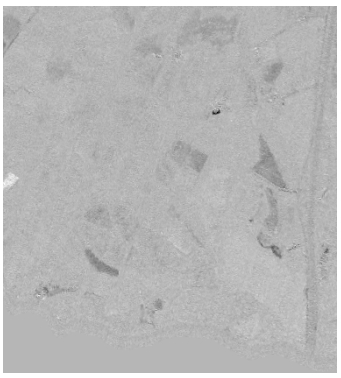
2.4.2.1 Positionnement multidimensionnel

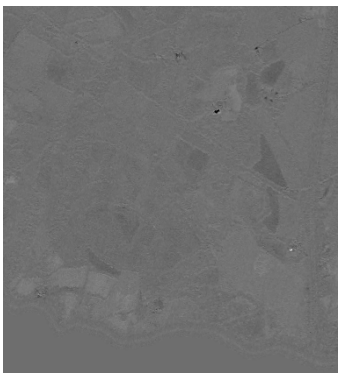
Le *positionnement multidimensionnel* (ou MDS) [50][131][172] regroupe plusieurs méthodes qui permettent d'estimer les coordonnées d'un ensemble d'individus dans un espace de dimension spécifiée, à partir de la connaissance des proximités entre ces objets. Les différents types de tableaux de proximité (distances, similarités, dominances, dissimilarités...) différencient ces méthodes (positionnement métrique, positionnement non-métrique etc...).
Le principe du positionnement multidimensionnel est donc de trouver une configuration de points dans un espace euclidien qui conservent au mieux les proximités initiales.
Le positionnement multidimensionnel est en quelque sorte une généralisation de l'ACP et permet, par exemple, de chercher la meilleure représentation euclidienne à partir de distances non-euclidiennes entre les individus.

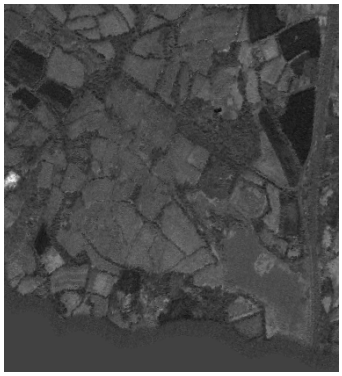
2.4.2.2 Cartes de Kohonen

Les cartes auto-organisatrices de Kohonen sont un outil connu et efficace de classification et de visualisation [126][127]. Il est également possible de les utiliser pour réduire la dimensionnalité d'un problème [191]. Le principe est de représenter les individus par


Composante indépendante 1


Composante indépendante 2


Composantes indépendante 3


Composante indépendante 4


Composante indépendante 5

FIG. 2.11 – Composantes Indépendantes de l'image hyperspectrale CASI présentée sur la figure 2.1

les coordonnées des centroïdes qui les représentent. La limitation de cette technique est qu'elle est essentiellement utilisée pour réduire la dimension à 2 ou 3. Or il est fréquent que la dimensionnalité intrinsèque d'un ensemble de données soit supérieure à 3. Les cartes de Kohonen sont conçues pour préserver la topologie entre l'espace d'entrée et celui de sortie. Deux points qui sont proches dans l'espace de départ sont projetés sur le même centroïde ou sur des centroïdes proches. Aussi, si la topologie est préservée, les distances ne le sont cependant pas, contrairement à des techniques comme le positionnement multidimensionnel.

2.4.2.3 Analyse en composantes curvilinéaires

L'Analyse en Composantes Curvilinéaires (ACC) [54] est une méthode neuronale dont le principe est de conserver au mieux la topologie en passant de l'espace de départ à l'espace d'arrivée. Elle consiste donc à minimiser un critère caractérisant la différence entre les deux topologies. Ce critère est défini par :

$$E_{ACC} = \frac{1}{2}\sum_{i}\sum_{j\neq i}(d(x_i,x_j) - d(x'_i,x'_j))^2 F(d(x'_i,x'_j)) \tag{2.10}$$

où :

- $d(x_i,x_j)$ est la distance euclidienne entre les points x_i et x_j dans l'espace de départ et $d(x'_i,x'_j)$) la distance euclidienne entre les projections de ces points dans l'espace d'arrivée
- $F : \mathbb{R}^+ \to [0,1]$ est une fonction décroissante.

F peut être choisie de la manière suivante [53] :

$$F_{\lambda(t)}(d(x'_i,x'_j)) = u(\lambda(t) - d(x'_i,x'_j)) \tag{2.11}$$

où $u(x)$ vaut 1 si $x \in \mathbb{R}^+$ et 0 sinon, et $\lambda : \mathbb{R}^+ \to \mathbb{R}^+$ une fonction "paramètre" décroissante.

La fonction F peut alors être minimisée à l'aide d'un algorithme de gradient. Pour réduire son coût de calcul, une *quantification vectorielle* [82] peut être effectuée préalablement. La quantification vectorielle peut elle-même être vue comme une méthode de réduction de dimension.

2.4.2.4 Autres méthodes

Il existe d'autres méthodes que nous ne détaillerons pas et qui n'ont pas été abordées dans cet exposé [4] :

- *Sammon's mapping* qui est très proche du positionnement multidimensionnel [55][165] ;

[4] Le but de cette section n'était pas d'établir une liste exhaustive des méthodes de réduction mais de présenter et de comparer brièvement les principales

- la *Régression non-linéaire* [5] qui fait appel à des notions proches de celles exposées ci-avant ;
- les *Courbes principales* [89] sont des courbes qui "passent au milieu" de l'ensemble des données initiales. Si on choisit ces courbes linéaires, on retombe sur le problème de l'ACP. Les courbes principales peuvent donc être vues comme une généralisation de l'ACP ;
- les Réseaux de densités [139] affectent une densité de probabilité a priori aux données puis utilisent des techniques d'apprentissage bayésien ;

2.5 Conclusion

Nous venons de voir dans ce chapitre que la multiplication du nombre de variables (et donc de la dimensionnalité) d'un ensemble de données, certes apportait une quantité d'information importante, mais engendrait également des problèmes théoriques et pratiques. Outre la redondance d'information et la lourdeur éventuelle des calculs, les grandes dimensions posent des problèmes lors de l'utilisation des méthodes d'analyse de données, en rendant par exemple difficiles -voire impossibles- les estimations de paramètres.

La connaissance de la dimension intrinsèque d'un échantillon, lorsqu'il est possible de l'obtenir, apporte une information intéressante sur ledit échantillon. Cette notion permet d'estimer la taille optimale de l'espace dans lequel on devrait projeter les données sans perdre d'information, et, par là même, de connaître la "dimension" de la redondance d'information. Autrement dit, la dimension intrinsèque informe du nombre de variables utiles (la notion d'utilité varie en fonction du type d'application envisagée).
Mais cette dimension intrinsèque n'a, le plus souvent, qu'un intérêt "consultatif". En effet, dans la plupart des problèmes nécessitant une réduction de dimension, la dimension de l'espace de projection est généralement donnée. Cette dimension est une contrainte du problème. Dans ce cas, connaître la dimension intrinsèque permet éventuellement d'apprécier la perte d'information. On peut donc considérer qu'on n'estime la dimension intrinsèque que lorsque l'utilisation d'une méthode de réduction l'exige.

Pour réduire la dimension du problème nous avons vu qu'il existait un grand nombre de méthodes. Parmi les méthodes linéaires, l'ACP et l'ACI sont sans doute les plus utilisées. L'ACP présente le premier avantage d'être simple et claire ce qui en fait une technique facile à utiliser pour des données issues de domaines très variés. D'autre part la possibilité de mesurer facilement la qualité des composantes et leur ordre naturel d'obtention par ordre décroissant d'information portée en sont deux autres avantages importants. Les résultats produits par l'ACP sont facilement interprétables.

Elle est très efficace mais la linéarité et l'orthogonalité des composantes obtenues peuvent être un obstacle à son utilisation systématique. L'ACI peut donner de meilleurs résultats dans le cas de données non Gaussiennes. Un de ses inconvénients est la complexité induite par l'optimisation de la fonction de contraste.

Les méthodes linéaires ne représentent qu'une petite proportion des techniques de réduction de dimensionnalité. Une grande partie des techniques actuelles sont des techniques non linéaires. Nous en avons présentées quelques unes parmi les plus classiques (positionnement multidimensionnel, cartes de Kohonen, etc.) et la plupart des autres méthodes en sont des dérivées ou des méthodes hybrides. Le principal inconvénient de cette famille de méthodes est la complexité de l'estimation nécessaire de certains paramètres qui oblige parfois de procéder à des approximations préléminaires.

Le choix de la méthode de réduction n'est pas toujours évident et résulte d'une étude des besoins, des contraintes et du type des caractéristiques des données. C'est toujours la conséquence d'un compromis entre "qualité" souhaitée de la projection et temps de calcul admissible. Cependant, le principe de parcimonie[5], suggère d'utiliser, lorsqu'elle est suffisante la méthode la plus simple. Outre les avantages cités précédemment, l'ACP s'avère être généralement un bon choix qui respecte ce principe de parcimonie, et c'est celui que nous avons fait dans le chapitre 3.

[5] «Le principe de parcimonie découle de la philosophie voulant que la nature soit parcimonieuse et que l'explication la plus simple soit souvent la meilleure. Dans l'analyse de parcimonie, on procédera en évaluant des milliers, voire des millions d'arbres (phylogénétiques) possibles pour déterminer l'arbre le plus court, c'est-à-dire le plus parcimonieux.» Bousquet, Jean, Des machines moléculaires à voyager dans le temps, Québec, revue Interface, Acfas, septembre et octobre 1995, p. 30.

Chapitre 3

Visualisation de données multidimensionnelles

Quand on ne sait rien, on peut tout de même trouver des choses, avec de l'imagination.
Boris Vian (Postface de "Les morts ont tous la même peau")

Sommaire

3.1 La visualisation de données 41
3.1.1 Méthodes de visualisation 42
3.1.2 Quelques méthodes spécifiques de visualisation 44
3.1.2.1 Matrices de Scatterplots 44
3.1.2.2 Courbes d'Andrew 45
3.1.2.3 Coordonnées parallèles 47
3.1.2.4 Glyphes, icônes et métaphores 49
3.1.2.5 Techniques Orientées-pixel 53
3.1.3 Logiciels, Packages 56
3.1.3.1 Logiciels et environnements de calculs numériques ou statistiques 56
3.1.3.2 Logiciels dédiés à la visualisation 56
3.1.4 Evaluation des techniques de visualisation 57
3.2 Problématique et cadre de notre travail 58
3.3 Visualisation par image couleur 63
3.3.1 Réduction de dimensionnalité 65
3.3.2 Calcul de la couleur d'une donnée de l'échantillon 66
3.3.3 Construction d'une image 68
3.3.4 Applications 70
3.3.4.1 Visualisation des classes sur des données simulées . . 71

3.3.4.2 Visualisation d'images multicomposantes 78
3.3.4.3 Visualisation de bases de données multidimensionnelles réelles 78
3.4 Visualisation dynamique 81
3.4.1 Concept 81
3.4.2 Application à la base de données IRIS 84
3.5 Discussion et conclusion 86

La visualisation graphique des données est une façon de présenter ces données de manière à ce que les structures sous-jacentes soient observables par l'oeil humain. Tufte écrit dans [187] "graphical excellence consists of complex ideas communicated with clarity, precision, and efficiency".

L'aspect "graphique" de la visualisation peut couvrir un large spectre de méthodes différentes comme les méthodes utilisant la 3D, la vidéo, le son ou même, faisant intervenir la perception tactile ou olfactive. Le type de visualisation graphique qui nous intéresse dans ce mémoire, est la visualisation utilisant un périphérique en 2 dimensions comme les écrans ou le support papier. Nous ne nous intéressons pas non plus aux méthodes faisant appel aux autres sens que la vue.

Les données que nous souhaitons visualiser sont des images multicomposantes, et, par généralisation, les données multidimensionnelles [109][185][153]. Nous nous intéressons à la finalité exploratoire de la visualisation et souhaitons l'utiliser pour explorer et analyser les images multicomposantes. En effet ce type de données particulières pose des problèmes de représentation. Visualiser une à une les composantes d'une image multidimensionnelle sur laquelle on ne sait rien a priori, ne permet pas d'avoir une vision globale et synthétique des informations contenues dans cette image. De même, si l'observation, une à une (voir deux à deux), des variables de données multidimensionnelles par des techniques classiques permet d'obtenir des informations sur ces descripteurs de façon partielle, il est difficile de se représenter les informations globalement. Autrement dit, l'observation des structures globales inhérentes aux données ne sont pas observables. Les méthodes traditionnelles échouent généralement à résoudre ce problème à cause du nombre trop grand de variables ou de la quantité trop élevée de données. Celles qui tendent à y parvenir, ne sont pas du tout adaptées aux images multicomposantes.

C'est cette lacune que nous avons tentée de combler en développant une méthode de visualisation utilisant une image statique en couleur, construite de manière non supervisée et utilisant une approche plus statistique que perceptuelle de la couleur.
Cette méthode est conçue pour la visualisation des images multicomposantes [14][16][17] mais nous l'avons également adaptée aux données multidimensionnelles quelconques

[19][15][21]. Elle permet de révéler les structures intrinsèques des données ou des images en fournissant une image couleur résumant l'information globale sous-jacente. En fournissant une image synthétique et immédiate des structures des données, elle apparaît être un outil intéressant comme première étape dans un processus exploratoire.
Nous avons également commencé à développer une méthode d'exploration dynamique de masses de données utilisant notre technique [18].

La première partie de ce chapitre sera consacrée à la présentation de méthodes de visualisation classiques existantes. Nous verrons par ailleurs qu'elles ne sont pas toujours adaptées aux ensembles de données de grande dimensionnalité et contenant beaucoup de points. Nous constaterons que ces méthodes sont inadaptées à la visualisation d'images multicomposantes. Nous présenterons alors notre méthode alternative qui est développée spécifiquement pour les images multicomposantes et qui, moyennant une étape supplémentaire, se révèle aussi très efficace pour exhiber les structures inhérentes aux données multidimensionnelles quelconques. Enfin, l'extension dynamique de notre méthode pour l'exploration de données sera proposée. Une discussion et une conclusion de nos contributions sont données à la fin de ce chapitre.

3.1 La visualisation de données

La plupart des auteurs qualifient de méthode de *visualisation* toute technique permettant de fournir une représentation graphique. Par exemple Card et col. [35] définissent la visualisation comme "the use of computer-supported, interactive, visual representations of data to amplify cognition". Le Petit Larousse Illustré [133] définit, en informatique, la visualisation comme une "présentation temporaire sur un écran, sous forme graphique ou alphanumérique, des résultats d'un traitement d'information" et rejoint donc la première définition.

Ces deux définitions permettent de séparer la visualisation en deux éléments : la représentation graphique proprement dite (i.e. la représentation sur le support choisi) et l'interprétation, l'activité cognitive, qu'elle permet de susciter chez l'utilisateur. L'utilisation de méthodes et outils informatiques permet de concilier ces deux aspects. En effet, l'outil informatique facilite le processus de visualisation par lequel le cerveau humain se construit une représentation mentale des données [178][195][40].

Ainsi la question implicitement soulevée par la conception de toute méthode de visualisation est : "Comment représenter graphiquement des données en préservant la signification intrinsèque et en fournissant un point de vue sur ces données ?". La réponse

n'est évidemment pas triviale. La recherche de solutions dépend de la nature des données, du type des informations à représenter et de leur utilisation [43]. De nombreuses idées ont souvent échoué lors de la mise en pratique. Tufte [187] et Bertin [9] ont à ce propos énuméré plusieurs représentations graphiques qui dénaturent l'information intrinsèque des données ou induisent de fausses idées. Tufte a alors proposé une liste de principes que devrait suivre toute méthode de visualisation :

1. "montrer" les données
2. éviter de dénaturer ou fausser ce que les données ont "à dire"
3. représenter de nombreuses données en peu de place
4. représenter les grands ensembles de données avec cohérence
5. faciliter les processus d'inférence, comme la comparaison de différents groupes de données
6. donner différentes perspectives sur les données, de l'aperçu global à l'observation de structures plus fines

Ces principes suggèrent implicitement plusieurs remarques classiques en visualisation :
- les données sont la plupart du temps multidimensionnelles alors que les représentations graphiques sur un écran ou sur papier sont présentées sur des surfaces 2D
- il est parfois nécessaire de représenter d'énormes ensembles de données alors que la quantité de données représentables sur un écran d'ordinateur ou sur une feuille est limitée
- le cerveau humain a la capacité remarquable de sélectionner, manipuler et réorganiser les données

Après avoir effectué un survol des méthodes de visualisation classiques et illustré leurs lacunes à représenter les données multidimensionnelles et les images multicomposantes, nous présenterons les particularités des données qui nous intéressent et les contraintes qui en découlent sur leur visualisation. Nous exposerons alors notre méthode de visualisation par l'image couleur ainsi qu'une extension possible pour l'exploration dynamique de masses de données. Enfin la dernière section conclura ce travail.

3.1.1 Méthodes de visualisation

Il existe un nombre très important de méthodes de visualisation (voir [90][3][145][31][188] ou encore [83]). On peut tout d'abord considérer que les méthodes statistiques descriptives sont aussi des méthodes de visualisation de données. Elles permettent en effet de décrire et d'explorer les bases de données multidimensionnelles. Comme l'expliquent Lebart et al. dans [135], "*Les techniques de statistique exploratoire multidimensionnelle mettent à profit [les] interfaces graphiques pour représenter, par exemple, les espaces factoriels et les arbres de classification : c'est la l'une de leurs fonctions iconographiques qui généralisent*

effectivement la statistique descriptive usuelle au cas de variables nombreuses". Il existe deux grandes familles de méthodes descriptives et exploratoires :

- les méthodes factorielles (Analyse en Composantes Principales, Analyse des Correspondances Multiples par exemple) qui produisent essentiellement des représentations des visualisations graphiques planes ou tridimensionnelles des données ;
- les méthodes de classification (voir chapitre 4 de ce mémoire) qui produisent des groupements des données en classes.

Ces méthodes ne sont pas spécifiquement dédiées à la visualisation de données et sont plutôt utilisées dans un cadre d'analyse statistique. Nous nous intéressons, dans ce chapitre, aux méthodes spécifiques de visualisation, c'est à dire à celles qui ont été conçues spécialement pour visualiser les données. Certaines de ces techniques utilisent d'ailleurs des méthodes exploratoires citées précédemment pour le traitement des données avant leur visualisation.

Les méthodes spécifiques de visualisation peuvent être classées de la manière suivante [163] :

- les méthodes géométriques ;
- les méthodes "orientées-pixel" ;
- les méthodes de représentation par "icônes", ou métaphores, ou glyphes ;
- les techniques hiérarchiques ;
- les techniques de distorsion ;
- les techniques de représentation basées sur les graphes.

Les méthodes géométriques regroupent des méthodes classiques comme les *coordonnées parallèles*[107][110][108][109], le *scatterplot*[22], *landscapes*, *hyperslices*... Les méthodes "orientées-pixel" consistent à associer un pixel à chaque donnée (ou groupe de données) et de visualiser le résultat sous la forme d'une image composée par ces pixels, rangés à l'aide de techniques de parcours (spaces filling curves) ou en mosaïque (par exemple). On se référera aux travaux de Keim [121] pour une liste quasi-exhaustive de ces méthodes. Les méthodes de visualisation par icônes, métaphores ou glyphes [161], ont pour principe d'associer à chaque individu, un élément graphique construit à partir des attributs de cet individu. Les techniques hiérarchiques s'intéressent habituellement aux données ayant une structure hiérarchique ou une structure de réseaux. Cependant des techniques hiérarchiques interactives ont été aussi développées pour la visualisation de données en grande dimension [203][204]. Elles utilisent souvent des représentations par arbres. Les techniques de distorsion proposent de distordre l'espace tri-dimensionnel pour permettre de représenter plus d'information (*Fish Eye View*, *Hyperbox*, *arbres hyperboliques*...). Enfin, les techniques basées sur les graphes utilisent des noeuds et des arcs pour représenter les données et requièrent donc beaucoup d'espace sur le périphérique d'affichage lorsque le nombre de données de l'ensemble à visualiser est élevé.

Nous allons maintenant dresser une galerie non-exhaustive de quelques unes de ces méthodes, parmi les plus utilisées et les plus classiques. Nous illustrerons ces techniques en les appliquant aux données de la base IRIS [12][150]. Cette base est composée de 150 données observées selon 4 attributs (et un 5ème indiquant la classe à laquelle appartient la donnée). Comme nous l'avons déjà dit à la section 2.1.1, cette base de données est très classique. La première classe de données est linéairement séparable des deux autres et ces deux autres classes ne le sont pas.

3.1.2 Quelques méthodes spécifiques de visualisation

Nous allons décrire brièvement et illustrer, dans cette section, quelques unes des méthodes de visualisation les plus utilisées. Nous présenterons successivement :

- trois méthodes géométriques (les matrices de scatterplots, les courbes d'Andrew et les coordonnées parallèles) qui seront utilisées sur la base IRIS (respectivement figures 3.13.2 et 3.4),
- le principe des glyphes et métaphores, appliqué ensuite à un sous-échantillon des données IRIS (voir figures 3.53.6,
- et le concept de visualisation orientée-pixels comme le définit D.A. Keim [120].

3.1.2.1 Matrices de Scatterplots

La méthode de *scatterplot* est une méthode classique de visualisation multivariée qui affiche les données selon deux attributs (i.e. selon deux variables). Pour les représentations de données de dimensionnalité supérieure à deux, on utilise les *matrices de scatterplot*.
Si on considère des données en dimension p, les matrices de scatterplots sont des objets qui représentent les scaterplot des projections selon les variables prises deux à deux. Autrement dit, la matrice de scatterplot représente les $p \times (p-1)$ scatterplots des projections. Chaque couple de variables est représenté par deux scatterplots montrant leurs relations.
La visualisation consiste à représenter les scatterplot sous forme d'un tableau où l'élément de la i^{eme} ligne et de la j^{eme} colonne représente le scatterplot de la i^{eme} variable en fonction de la j^{eme}. Le scatterplot symétrique, de la j^{eme} ligne et de la i^{eme} colonne représente la même relation en interchangeant les deux axes.
L'exemple de la figure 3.1 montre la matrice de scatterplots de la base de données IRIS, obtenue avec Matlab. Sur cette représentation, les éléments diagonaux -qui d'habitude contiennent les "noms" éventuels des variables- contiennent les histogrammes des variables. Autrement dit, le i^{eme} élément de la diagonale représente l'histogramme des données selon la i^{eme} variable.
On observe bien, sur cette représentation qu'une classe se détache des deux autres mais

la séparation des deux autres est impossible. Par ailleurs la lecture nécessite un certain effort compte tenu du nombre de variables. On verra, dans la section 3.2, que cette lecture devient impossible quand ce nombre de variables augmente encore.

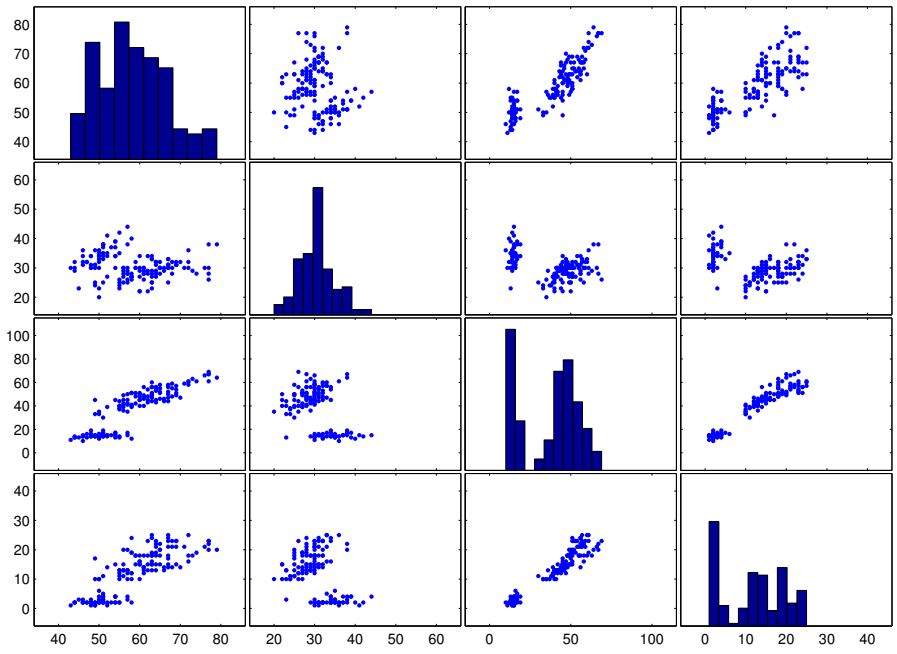


FIG. 3.1 – *Matrice de Scatterplots* de la base de données IRIS

3.1.2.2 Courbes d'Andrew

D. Andrews a introduit cette méthode en 1972. Le principe est de représenter chaque individu de l'échantillon à visualiser par une fonction périodique $f_{x_i}(t)$ (si on appelle x_i l'individu) dont les coefficients sont les attributs de cet individu (i.e. $x_i^1, x_i^2, \ldots, x_i^p$). La figure 3.2 représente cette fonction pour l'ensemble des données IRIS. Chaque courbe représente un individu.

On observe bien, sur ce graphe, que deux "classes" de courbes se distinguent. La visualisation de cette séparation est assez immédiate. La deuxième séparation est cependant invisible. On verra (section 3.2) que cette visualisation est inefficace lorsque le nombre de données augmente.

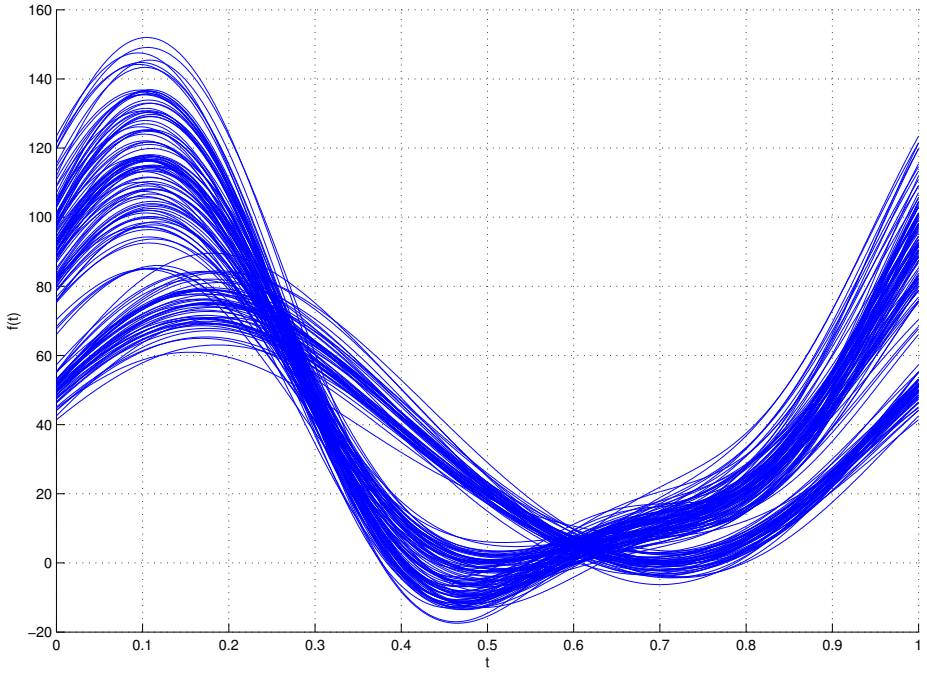


FIG. 3.2 – Courbes d'Andrew pour la base de données IRIS

3.1.2.3 Coordonnées parallèles

Le système de coordonnées parallèles est particulièrement adapté pour les données multidimensionnelles. Chaque individu est représenté par une ligne brisée à travers les coordonnées parallèles (comme le montre la figure 3.3). Les variables sont représentées parallèlement, se succédant horizontalement. Ce schéma montre la représentation du point $x = (3, 5, 3, 4, 3) \in \mathbb{R}^5$ en coordonnées parallèles. Le premier attribut de ce point vaut 3 et le second vaut 5. Le premier attribut est caractérisé par la coordonnées 1 et le second par 2. Le tracé commence donc par un segment reliant le point d'abscisse 1 et d'ordonnée 3 au point d'abscisse 2 et d'ordonnée 5. On construit ensuite le segment reliant le second attribut au troisième et ainsi de suite.
Ce type de tracé permet de voir les relations entre les variables (la corrélation par exemple). Il présente entre autre l'avantage de pouvoir facilement comparer les individus deux à deux. La figure 3.4 représente la visualisation en coordonnées parallèles de la base de données IRIS.

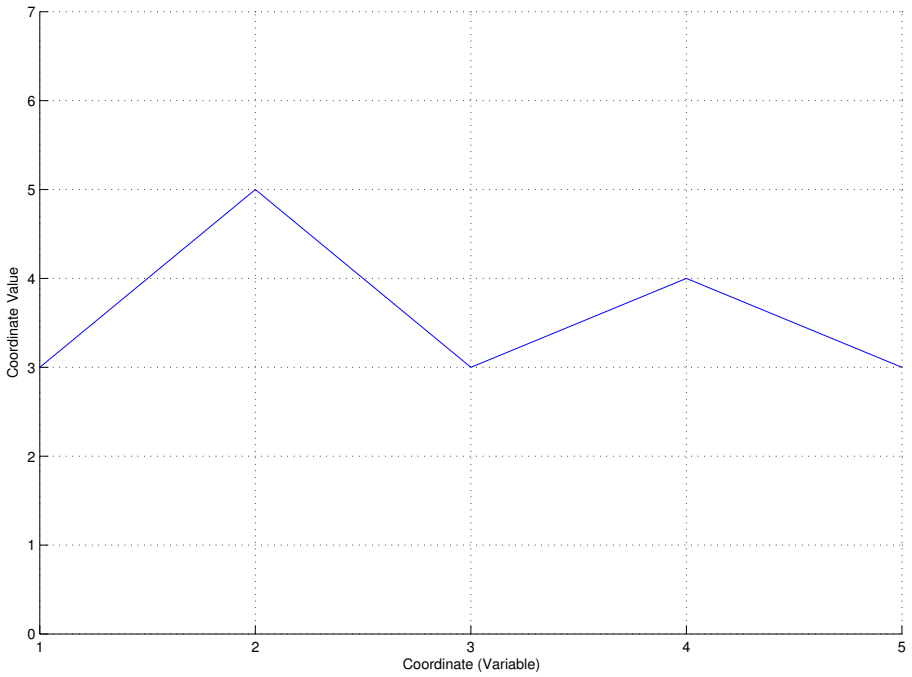


FIG. 3.3 – Représentation en coordonnées parallèles du point en dimension 5, $x = (3, 5, 3, 4, 3) \in \mathbb{R}^5$.

Sur cet exemple (figure 3.4), on parvient à distinguer deux ensembles bien "séparés"

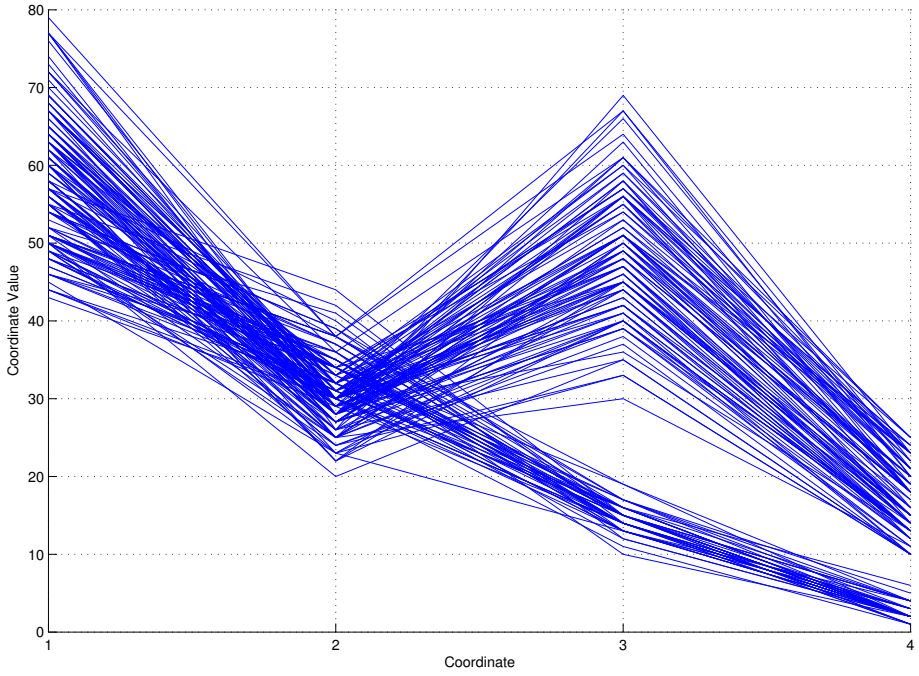


FIG. 3.4 – Coordonnées parallèles de la base de données IRIS

de données. Or la base de données IRIS contient 3 "vraies" classes. Une classe reste donc masquée avec cette visualisation.

Remarque : l'ordre des variables est important dans ce type de représentation, et la lecture peut être très perturbée s'il n'est pas adéquat.

Le nombre de points à afficher est également une limitation de cette représentation (section 3.2).

3.1.2.4 Glyphes, icônes et métaphores

Les représentations sous forme de glyphes, d'icones ou de métaphores consistent à représenter chaque individu de l'échantillon comme un élément graphique dont les caractéristiques dépendent des attributs des données. Picket et Grinstein [158] ont présenté une technique de visualisation appelée *stick figure icon* et Beddow une représentation dite par *Autoglyph*. Les figures 3.5 et 3.6 représentent une visualisation de 15 données extraites de la base IRIS utilisant deux types de *glyphes*. Dans le premier cas, les glyphes sont des "étoiles" dans lesquelles chaque segment représente un attribut, et la longueur de ce segment est proportionnelle à la valeur de l'attribut correspondant. Dans le second cas, les symboles utilisés sont les visages de Chernoff (*Chernoff faces*). Pour représenter les différents attributs, le visage possède des caractéristiques variables.
Exemples de caractéristiques :

- forme de la tête
- taille des yeux
- forme des yeux
- espacement entre les yeux
- centrage des yeux
- taille de la pupille
- taille du nez
- forme de la bouche
- taille de la bouche
- ouverture de la bouche

La figure 3.7 contient la totalité de l'échantillon IRIS représenté à l'aide des deux types de glyphes sus-cités. On imagine alors assez bien la difficulté à lire une telle représentation lorsque le nombre de données à afficher est assez grand.

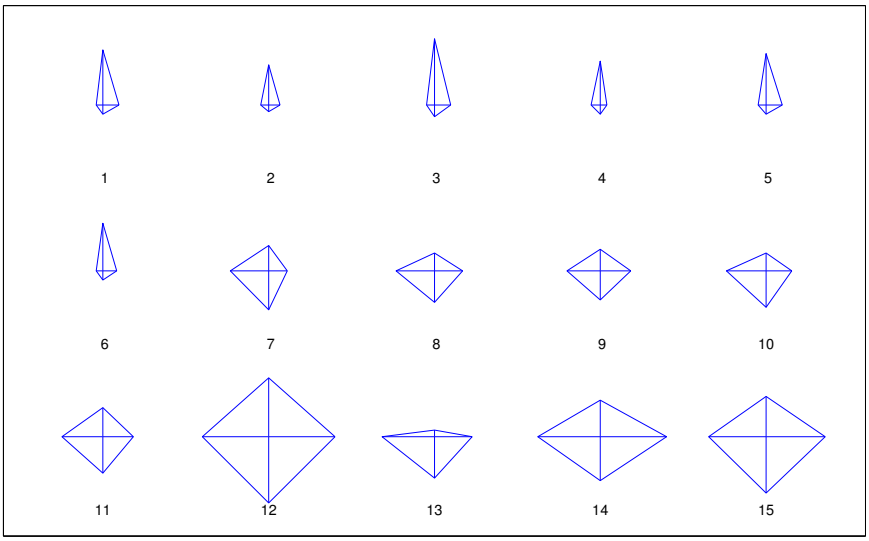


FIG. 3.5 – Visualisation par glyphes-étoiles (*Glyphs 'Star'*) de 15 données extraites de la base de données IRIS

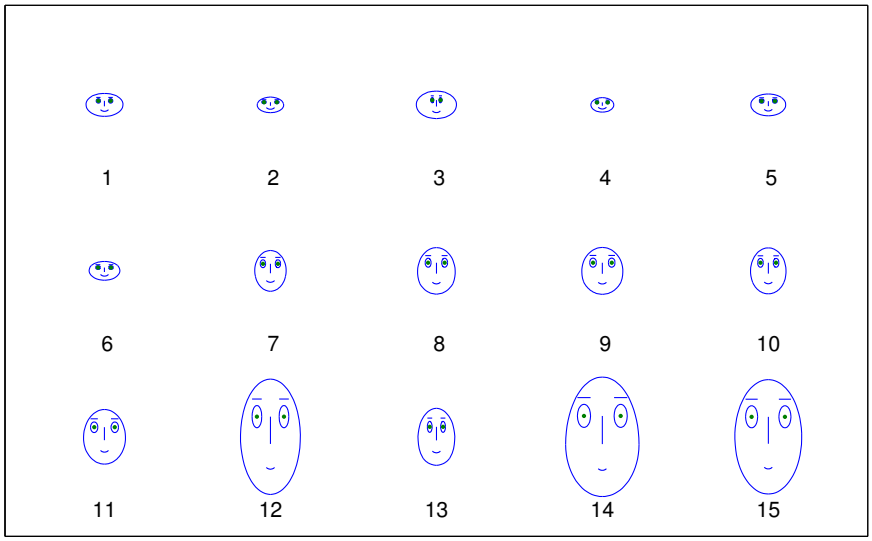


FIG. 3.6 – Visualisation par glyphes-visages (*Glyphs 'Face'*) des même 15 données extraites de la base de données IRIS

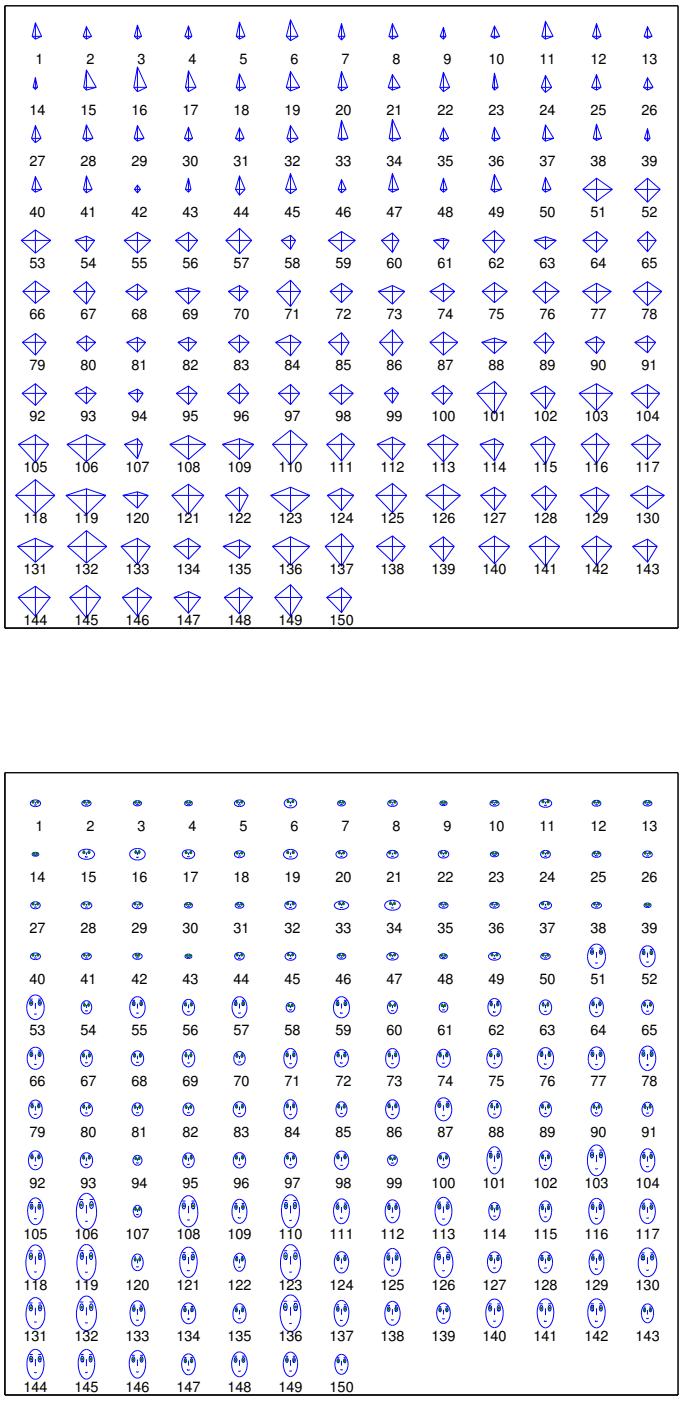


FIG. 3.7 – Visualisation par *Glyphs 'Star' et 'Face'* de la base de données IRIS

3.1.2.5 Techniques Orientées-pixel

Le principe des techniques de visualisation orientées-pixel, très largement étudiées et développées par D. A. Keim et H.-P. Kriegel [122][120][121], est d'associer un pixel couleur à chaque élément de la base de données à visualiser, puis d'arranger les pixels dans une image pour les visualiser. Les techniques de ce type comportent donc deux étapes importantes :

1. associer une couleur à chaque donnée (étape appelée *color mapping*)
2. organiser spatialement les pixels obtenus à l'étape précédente, afin de visualiser les données sous forme d'une image

On peut dors et déjà faire une première remarque commune à toutes ces techniques : puisque l'on fait correspondre un pixel à une donnée, le nombre de données représentables n'est donc limité que par la capacité d'affichage du support utilisé.
Par ailleurs, de nombreuses techniques orientées-pixel utilisent un partitionnement de la fenêtre en sous-fenêtres. Ce principe permet de représenter des données multidimensionnelles. Supposons par exemple que l'on souhaite représenter des données en dimension p. Il s'agit alors de diviser l'écran d'affichage en m sous-fenêtres (une pour chaque dimension), ou $m+1$ sous-fenêtres dans le cadres de visualisations dépendant de requêtes. Dans ce cas en effet, la sous-fenêtre supplémentaire est consacrée aux distances globales. Le figure 3.8 présente deux exemples de découpages, l'un en sous-fenêtres rectangulaires et l'autre en secteurs.

Nous allons maintenant voir comment sont réalisées les deux tâches constituant ce type de méthode de visualisation.

***Color mapping* ou affectation de couleurs** Cette première étape consiste donc à associer une couleur à chaque donnée de l'échantillon à visualiser. L'intérêt d'utiliser de la couleur plutôt que des niveaux de gris est de pouvoir représenter plus de différences entre les données [97]. Dans les travaux de Keim, ce problème consiste à trouver une échelle de couleur aussi perceptuellement efficace que possible. On trouvera, dans [121], plusieurs réflexions sur le choix de cette échelle de couleur.

Arrangement des pixels La seconde question concerne la façon d'organiser spatialement les pixels pour l'affichage. Comment arranger les pixels dans l'image ? Cette étape est cruciale et doit permettre d'exhiber les informations apportées par la couleur et donc de dévoiler les structures des données. Elle nécessite d'avoir :

- un ordre sur les données -donc sur les pixels- (qui est une forme de projection unidimensionnelle)
- une technique de remplissage de la fenêtre d'affichage à l'aide des pixels ainsi ordonnés

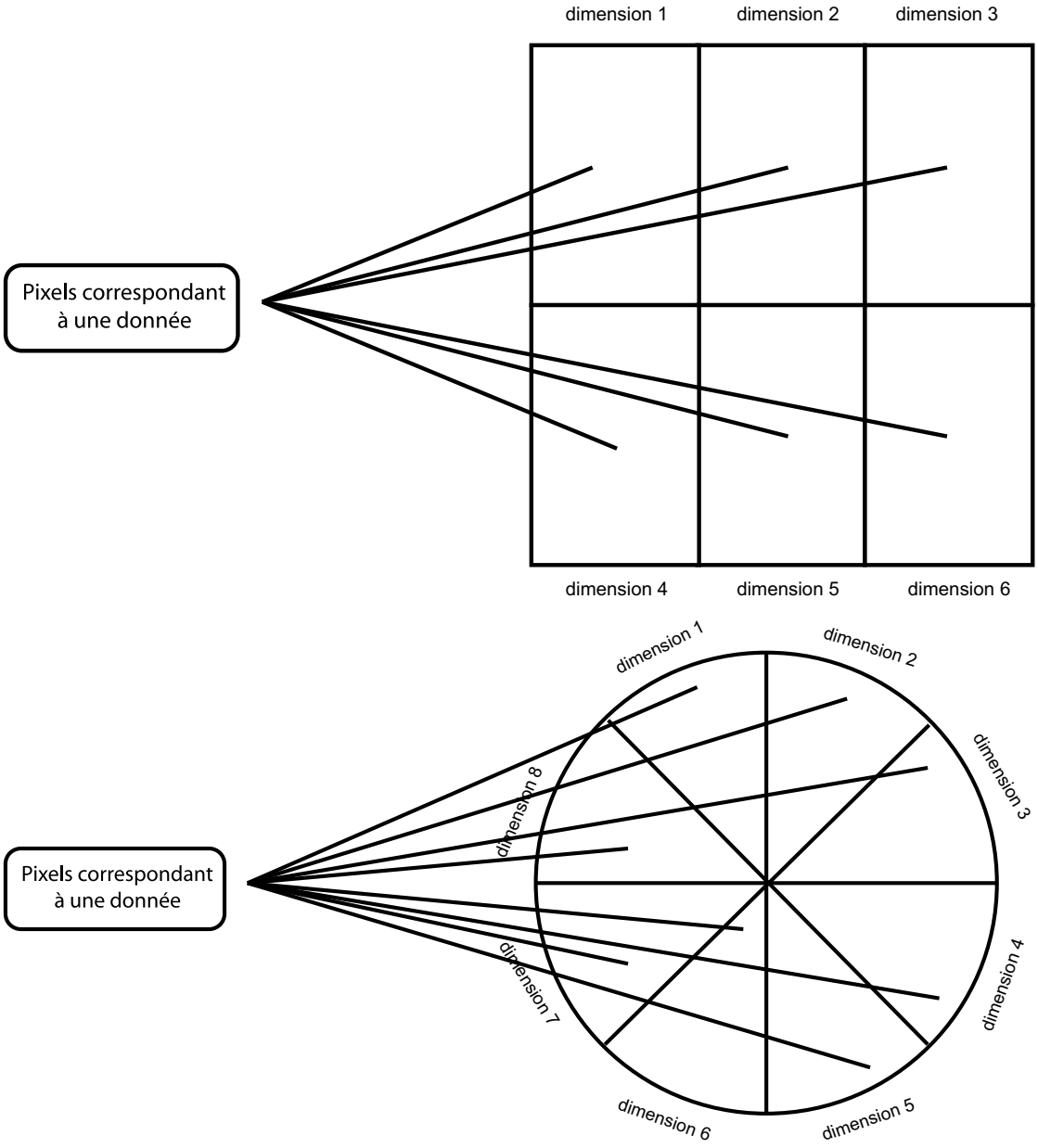


FIG. 3.8 – Exemple de découpages classiques en sous-fenêtres. L'exemple du haut concerne des données en dimension 6 tandis que celui du bas concerne les données en dimension 8.

Certaines données ont un ordre implicite de par leur nature, comme par exemple les données temporelles (qui sont naturellement ordonnées selon le paramètre "temps". Pour les autres types de donner il est nécessaire de trier les données. Ce tri peut être fait en fonction de critères particuliers, de requêtes, imposés par le contexte, le problème ou l'utilisateur (triés selon un attribut particulier par exemple). Il est également possible de trier les données en fonction de critères dépendants des données (les données peuvent par exemple être triées selon la première composante principale obtenue par l'ACP du tableau de données).

Lorsque les données sont triées, il s'agit alors de trouver une application bijective entre les données ordonnées et une fenêtre d'affichage de taille fixée. On peut considérer que les données triées sont rangées, indexées, sur une courbe unidimensionnelle, et qu'il faut remplir, à l'aide de cette courbe une image ou une fenêtre d'affichage. Une propriété souhaitable de cette application serait de faire en sorte que deux données qui sont proches (au sens de l'ordre sur ces données) soient proches dans l'affichage.
Une bonne solution à ce problème d'optimisation est fournie par les courbes de Hilbert-Peano [156][98][179]. Ces courbes remplissent un espace discret de dimension 2 à l'aide d'une "courbe" de dimension 1. Les plus classiques sont les courbes dites en "U" et en "Z" (Figure 3.9). Ces courbes fractales se déterminent facilement à l'aide d'algorithmes récursifs par exemple [32].

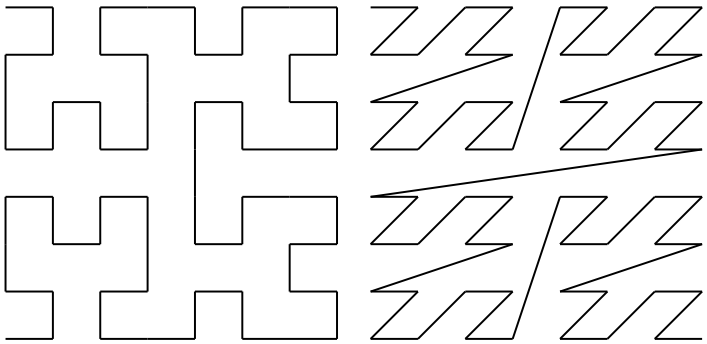


FIG. 3.9 – Courbe de Peano-Hilbert en "U" (gauche) et en "Z" (droite). Ces courbes remplissent une grille carrée de 8 points de coté.

Ces courbes présentent de nombreux avantages sur les techniques "naïves" comme celle consistant à remplir de gauche à droite et de haut en bas (balayage) [175][176][177][168] :

- Elles présentent la propriété souhaitable suivante : deux points qui sont proches sur la courbe unidimensionnelle, le sont également dans l'espace rempli par cette courbe
- Ces courbes préservent donc les propriétés de clustering

– Elles permettent d'éviter les problèmes liés aux techniques d'affichage classiques (distorsion, effet de Moiré etc...)

Après avoir passé en revue quelques techniques spécifiques classiques, nous allons citer quelques exemples de packages et logiciels intégrant, notamment, ces méthodes. On distinguera les logiciels statistiques ou de calcul assez "généraux" des logiciels spécifiques de visualisation.

3.1.3 Logiciels, Packages

Il existe de nombreux logiciels et packages implémentant certaines des techniques évoquées dans cette partie. On peut citer par exemple :

3.1.3.1 Logiciels et environnements de calculs numériques ou statistiques

– les packages MATLAB pour le tracé des coordonnées parallèles, Andrew's curves, glyphs...
– le logiciel/langage de calcul statistique `R` possède de nombreux packages incluant les méthodes classiques de visualisation (on peut également citer son "cousin", le logiciel SPlus).
– Mathematica est un environnement de calcul formel et numérique qui permet également de visualiser des informations en 2D ou 3D avec des techniques de surfaces, de cartes, de scatterplots. Cet outil est plus particulièrement conçu pour les objets mathématiques et pas les bases de données.
– SAS est un logiciel de Statistique qui possède de nombreux modules supplémentaires et dépasse largement les seules méthodes statistiques. Outre tous les types de visualisation d'analyse de données et statistiques classiques, SAS dispose d'un module, JMP, étendant ses possibilités de visualisation de données multidimensionnelles.

3.1.3.2 Logiciels dédiés à la visualisation

– le logiciel VisDB développé par Keim et col. qui implémente notamment des méthodes orientées-pixel ;
– IRIS Explorer est un produit de Silicon Graphics Inc. qui fournit de nombreuses librairies et un environnement de programmation pour le développement de modules de visualisation ;
– VisuLab est un outil de l' Institute for Scientific Computing of ETH Zurich qui permet de visualiser simultanément, sur la même fenêtre, les données multidimensionnelles de différentes manières (Scatterplots, coordonnées parallèles, Andrew's curves etc... ;

- AVS est un produit commercial de Advanced Visual System Inc. qui est composé de 5 applications interactives de visualisation : Geometric Viewer, Image Viewer, Graph Viewer, Data Viewer et Network Editor ;
- XmdvTool est un outil intéressant. Gratuit, il permet à l'utilisateur d'explorer les données multidimensionnelles de plusieurs façons différentes, principalement des méthodes de projection (scatterplots, glyphs, coordonnées parallèles ;
- Le logiciel AMIRA est plutôt orienté "synthèse d'images" et permet de visualiser des données en $3D$.

Cette liste n'est évidemment pas exhaustive, il existe de nombreux autres logiciels de visualisation. Ceux de cette liste sont parmi les plus connus et/ou les plus utilisés. Le choix peut sembler large pourtant il est à noter qu'il existe peu d'environnements consacrés spécifiquement au traitement d'images.

Nous venons donc de voir qu'il existait une grande variété de techniques de visualisation et de logiciels qui les implémentent. Il convient maintenant d'étudier les différences entre ces techniques et de les évaluer afin de permettre de faire un choix en fonction du problème posé.

3.1.4 Evaluation des techniques de visualisation

Nous allons donc, dans cette section, donner des éléments de comparaison et d'évaluation de toutes ces méthodes afin de permettre de savoir quand utiliser quelle technique. Autrement dit, cette section doit nous aider à choisir la méthode de visualisation la plus adéquate.

Tout d'abord notons que toutes ces techniques ne sont pas simples d'utilisation. En effet, certaines (les techniques géométriques) produisent des affichages d'une complexité telle qu'elles nécessitent une analyse poussée pour permettre la compréhension.

On trouve, dans [123], une comparaison poussée de toutes les techniques présentées. La table 3.1 présente un comparatif du comportement de quelques techniques ou familles de techniques dans le cadre de différentes caractéristiques des données ou structures des données.

Pour comparer -et donc choisir- les méthodes de visualisation, il est nécessaire d'identifier le but de la visualisation et d'examiner les caractéristiques du type des données. En effet la visualisation doit permettre de mettre en valeurs les structures inhérentes aux données comme les corrélations, les dépendances fonctionnelles ou les classes.
Avant de choisir une méthode de visualisation il convient d'analyser les points suivants :

Le problème : il s'agit d'essayer de comprendre ce qui doit être présenté, trouvé ou démontré.

La nature des données : les données peuvent être de type numérique, ordinal, ou catégoriel.

Dimensionnalité : le nombre de dimensions à représenter est un critère important. Si les données sont en dimension inférieure ou égale à 3, le choix est facile, mais en dimension très élevée, le nombre des techniques qui restent efficaces est faible.

Structure des données : la structure des données peut être linéaire, temporelle, spatiale, hiérarchique ou les données peuvent avoir une structure de réseau. Le traitement est alors évidemment différent selon les cas.

Nombre de données : le nombre de données est également une caractéristique à prendre en compte (comme nous le verrons plus loin).

Tous ces critères suggèrent donc naturellement d'orienter les choix vers des techniques différentes selon les cas (exemple : voir table 3.1).

Nous venons de voir les critères à retenir pour choisir une méthode de visualisation. Comme nous l'avons déjà dit, le leitmotiv de notre travail est le traitement des images multicomposantes. Nous allons donc présenter dans la partie suivante les contraintes liées à ce type de données et constater que les méthodes existantes présentées dans cette section ne conviennent pas.

3.2 Problématique et cadre de notre travail

Nous nous sommes intéressés, dans ce travail, à la visualisation d'ensembles de données multidimensionnelles contenant un grand nombre d'individus. Les images multicomposantes constituent des ensembles de données particuliers :

- dont le nombre d'attributs peut être élevé
- dont le nombre d'individus est très important

Or, lorsque la dimensionnalité et le nombre de données augmentent, les techniques traditionnelles présentées avant se révèlent inefficaces et deviennent rapidement illisibles, rendant l'observation et l'interprétation difficile voire impossible (Figures 3.10, 3.11, 3.12).

Par ailleurs, toutes les techniques citées ne sont pas adaptées à la visualisation d'images multicomposantes. En effet, dans toutes ces techniques, l'information visuelle -apportée par la structure d'image- de chacune des composantes est perdue lors de la visualisation. L'information de localisation des individus n'est pas exploitée visuellement. Les images multicomposantes nécessitent donc des outils spécifiques.

Enfin nous avons choisi de développer une méthode qui ne nécessite pas d'apprentissage, ni de connaissances a priori sur les données. Nous avons donc développé une

		Tâches, Particularités						
		1	2	3	4	5	6	7
Techniques	Matr. de scatterplot	++	++	+	+	-	0	++
géométriques	Landscapes	+	+	-	0	0	+	+
	Projections	++	++	+	+	-	0	+
	Hyperslice	+	+	+	+	-	0	0
	Coord. parall.	0	++	++	-	0	-	0
Icones et	Stick Figures	0	0	+	-	-	-	0
métaphores	Shape coding	0	-	++	+	-	+	-
	Icones couleur	0	-	++	+	-	+	-
Orientées	query ind.	+	+	++	++	-	++	+
pixel	query dep.	+	+	++	++	-	++	-
Techniques	Dim. Stack.	+	+	0	0	++	0	0
hiérarchiques	Treemap	+	0		0	++		0
	Cone Trees	+	+	0	+	0	+	+
Techniques	basic graphs	0	0	-	+	0	0	+
de graphes	Spec. graphs	++	+	-	+	0	+	+

où les caractéristiques, représentées en colonnes, sont :

1 Clustering
2 Point chaud multivarié
3 Grand nombre de variables
4 Grand nombre d'individus
5 Données catégorielles
6 Superposition visuelle
7 Courbes d'apprentissage

TAB. 3.1 – Comparaison des techniques de visualisation selon D. A. Keim en fonction des particularités des données et du contexte d'utilisation (++ : très bien, + : bien, 0 : neutre, - : mauvais)

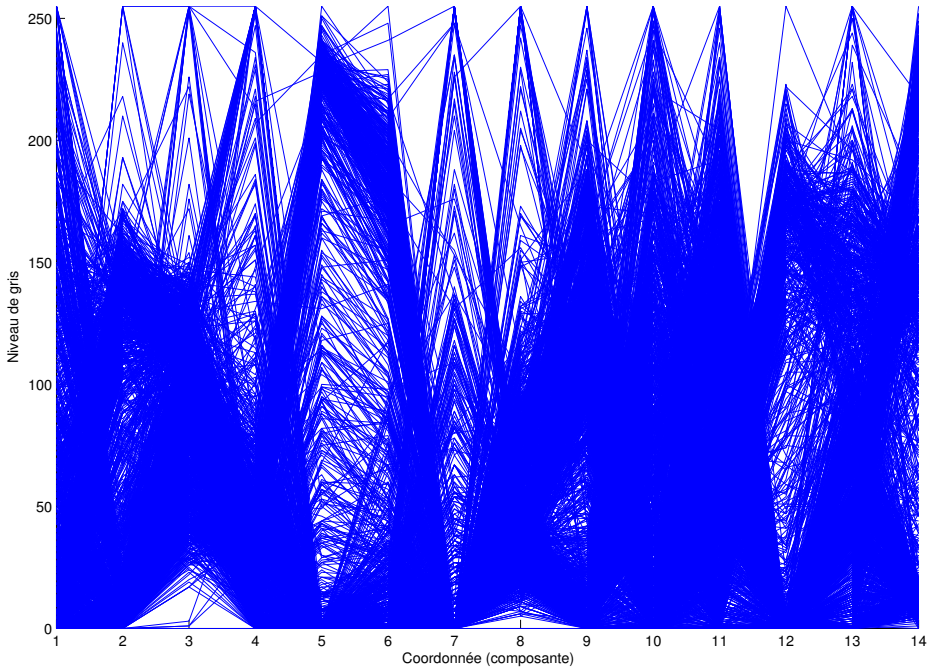


FIG. 3.10 – Coordonnées parallèles des données correspondant à l'image multicomposante de fluorescence X. La lecture est rendue quasi-impossible par le nombre de points élevé de l'échantillon (1560 individus)
et par la difficulté à ordonner les composantes.

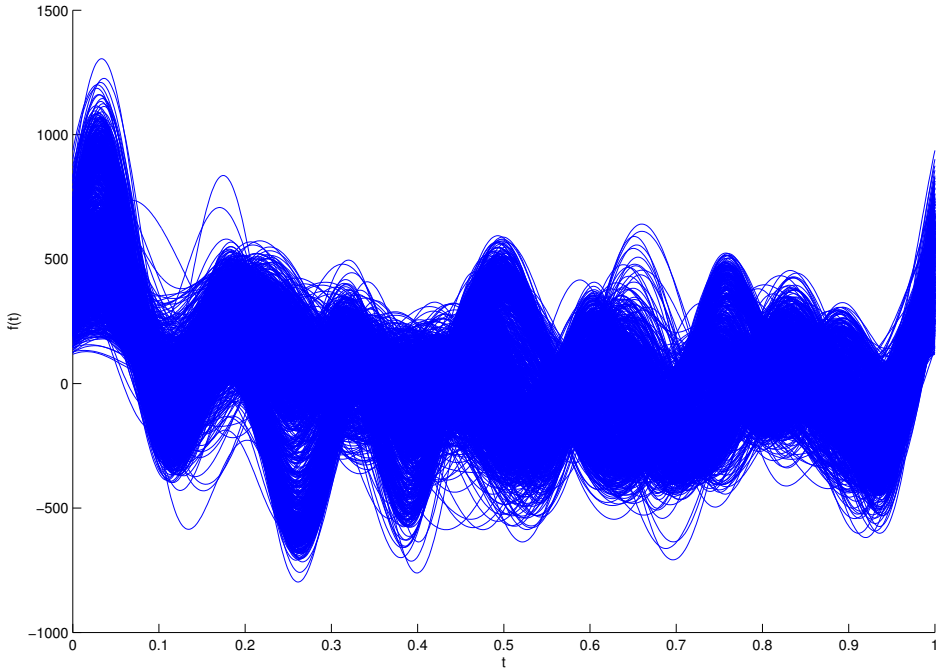


FIG. 3.11 – Visualisation par Andrews Plot des données correspondant à l'image multi-composante de fluorescence X. La représentation est à nouveau rendue illisible à cause de nombre de points élevé de l'échantillon (1560 individus)

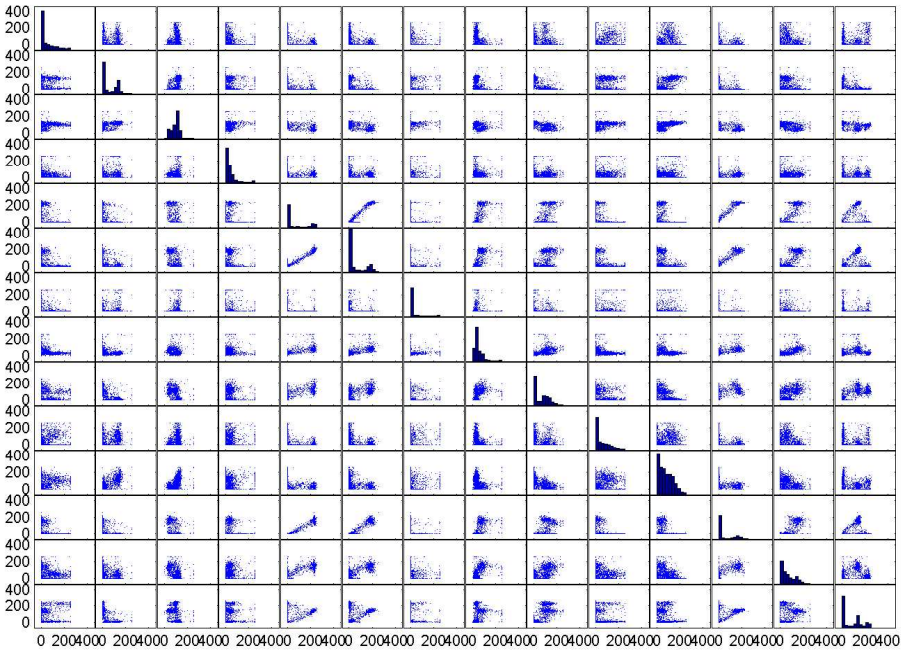


FIG. 3.12 – Visualisation par Matrice de Scatterplots des données correspondant à l'image multicomposante de fluorescence X. La dimensionnalité élevée du problème rend la visualisation difficile à cause du nombre élevé de Scatterplots à afficher. Le nombre de points élevé n'arrange rien.

méthode de visualisation efficace pour les données multidimensionnelles et particulièrement bien adaptée aux images multicomposantes aussi “automatique” que possible.

3.3 Une nouvelle approche : visualisation non supervisée de données par l’image couleur

Comme nous venons de le voir dans la section précédente et dans le chapitre d’avant, l’analyse des données multidimensionnelles et la visualisation de ces données deviennent particulièrement complexes quand la dimension et le nombre des données augmentent. L’objectif de cette partie est de présenter une nouvelle méthode de visualisation reposant sur l’utilisation de la couleur pour définir une représentation plane des données multidimensionnelles. La technique que nous proposons utilisera l’image numérique et la couleur pour visualiser un ensemble de données multidimensionnelles appartenant à un espace de dimension supérieure ou égale à trois. Si cette dimension est strictement plus grande que trois, une réduction de dimension sera nécessaire. Le choix des couleurs utilisées sera objectif et non supervisé. La technique proposée sera particulièrement adaptée pour des échantillons d’effectifs relativement importants (plus de 10.000 données) mais elle restera utilisable pour des échantillons de taille plus modeste. Elle est présentée pour des données quantitatives quelconques mais se voit simplifiée lorsque l’échantillon est constitué de l’ensemble des pixels d’une image multicomposante.

La visualisation de grands échantillons de données par des techniques orientées-pixels n’est pas une approche nouvelle (comme nous l’avons vu dans la partie 3.1.2.5). Elle nécessite de définir une correspondance bijective entre l’ensemble de données et les pixels de l’image. La méthode classique de construction de l’image consiste à classer les données dans un certain ordre, l’ensemble des données forme alors une ligne de pixels successifs (i.e. de données successives), il faut ensuite définir le parcours que doit suivre cette ligne pour remplir toute l’image. Parmi les techniques de remplissage d’une image par une courbe, le parcours de Peano est l’un des plus connus (voir [147] et la première partie de ce chapitre [Figure 3.9]). C’est cette technique que nous avons retenue. Lorsque l’on souhaite utiliser l’image couleur, il faut aussi affecter une couleur (c’est-à-dire un triplet (R, V, B), Rouge Vert Bleu) à chaque pixel, donc à chaque donnée.

La méthode que nous avons développée est une méthode orientée-pixel utilisant la couleur [21]. Elle consiste à affecter, à chaque donnée, un pixel couleur puis à organiser ces pixels couleurs dans une image [19]. Dans le cas d’images multicomposantes, cette phase d’organisation spatiale des pixels est facultative mais peut être intéressante (comme nous le verrons dans la suite). Autrement dit, on associe à chaque pixel multidimensionnel un pixel couleur ayant les mêmes coordonnées dans l’image produite. On obtient donc, dans

ce cas, une image couleur à partir l'ensemble des images en niveau de gris constituant l'image multicomposante. L'organisation spatiale de l'image générée est la même que celle des images composantes [17].
Notre technique fournit ainsi une vision synthétique et immédiate des structures inhérentes aux données [18]. L'information contenue dans l'ensemble des données est exhibée et visible instantanément grâce à la couleur. Comme nous le verrons plus loin dans cette section, elle permet de mettre en évidence des classes de données. Elle constitue donc un outil intéressant comme étape préliminaire dans un processus exploratoire.

Cette approche que nous proposons se décompose en deux (ou trois) étapes :

1. la première consiste à réduire la dimension des données à trois par projection dans un espace de dimension trois, c'est une étape de réduction de dimension.
2. la seconde permet d'affecter à chaque triplet (X, Y, Z) qui correspond à une donnée projetée, un nouveau triplet (R, V, B) qui sera la couleur affectée à la donnée initiale.
3. la troisième n'est pas systématique et permet, dans le cas de données non spatialisée, d'arranger les pixels obtenus à l'étape 2 dans une image résultat.

Le choix des couleurs est souvent très subjectif et a pour but de mettre en évidence les informations que l'on pense être pertinentes pour l'observateur [92][93][]. Dans notre travail, nous proposons une approche plus objective et non supervisée basée sur une méthode statistique. Il s'agit d'utiliser la transformée inverse de celle de Otha, Kanade et Sakai [151].
La transformation de Ohta, Kanade et Sakai appliquée aux triplets (R, V, B) d'une image couleur permet d'obtenir de nouveaux triplets (C_1, C_2, C_3) qui approxime la transformation de Karhunen-Loève (TKL). Dans la situation inverse de celle étudiée par Ohta et al., si nous avons trois composantes obtenues par la TKL sur un échantillon de données, nous proposons d'utiliser la transformation inverse de celle de Otha et al. pour approximer la couleur. Appliquée à un triplet (C_1, C_2, C_3), cette transformation inverse approximera la couleur (R, V, B) de la donnée à l'origine de (C_1, C_2, C_3). Cette approximation nécessite d'abord d'avoir calculé les trois premières composantes (C_1, C_2, C_3) de l'échantillon de données, ensuite la transformée inverse de Ohta et al. donnera la couleur à affecter à chaque donnée.
Si on considère un pixel multidimensionnel caractérisé par le n-uplet $(x_1, x_2, \ldots, x_n)$, le processus d'attribution d'une couleur à ce pixel par notre méthode peut se schématiser sous la forme suivante :

$$(x_1, x_2, \ldots, x_n) \xrightarrow{1} (C_1, C_2, C_3) \xrightarrow{2} (R, G, B)$$

La dernière étape, surtout utile dans le cas de données non spatialisées, consiste à organiser spatialement les pixels couleurs (associés aux données dans la première étape) dans

une image. On utilise pour ça une courbe de remplissage de Peano en U.

Dans la section suivante (3.3.1), nous montrerons la nécessité de réduire la dimensionnalité ($\xrightarrow{1}$) , dans le cadre de la visualisation, et indépendamment des contingences liées à notre méthode de visualisation. La partie 3.3.2 exposera notre méthode d'affectation des couleurs à un ensemble de données projetées dans un espace de dimension trois ($\xrightarrow{2}$). Ensuite, la partie 3.3.3 décrira la construction de l'image présentant l'ensemble des données. Enfin, la partie 3.3.4 présentera des applications et validations de cette nouvelle méthode.

3.3.1 Réduction de dimensionnalité

L'analyse de données multidimensionnelles nécessite une réduction de dimensionnalité pour des raisons pratiques liées aux représentations des données mais aussi pour des raisons théoriques que nous rappellerons brièvement.

L'être humain peut concevoir un espace de dimension trois et les progrès des systèmes de visualisation permettent des présentations et des manipulations d'objets 3D. L'exploration d'espaces de dimension supérieure à trois nécessite l'introduction de métaphores comme le temps pour produire des visualisations dynamiques donnant des indications sur une quatrième dimension. L'utilisation d'icônes, de graphes, de la couleur ou de textures permet aussi d'augmenter les possibilités d'exploration d'espaces de dimension supérieure à trois [91]. Cependant la dimension trois reste une limitation naturelle de la perception humaine de l'espace. Pour cette raison pratique, on peut considérer qu'une réduction de dimensionnalité à trois est optimale par rapport à la perception humaine de l'espace mais cette raison pratique n'est pas la seule (voir section 2.2 consacrée aux grandes dimensions).

Dans notre méthode nous utilisons une approche très classique, simple et généralement efficace de la réduction de dimensionnalité : nous conservons les trois premières composantes générées par une Analyse en Composantes Principales (ACP) [159]. Pour rappel, le principe de l'ACP (voir section 2.4.1.1) est de projeter les données initiales dans un sous espace de dimension réduite k (ici $k = 3$). Ce sous-espace est optimisé pour maximiser l'inertie du nuage des données projetées. Le sous-espace est engendré par les k vecteurs propres correspondants aux k plus grandes valeurs propres de la matrice de covariance. Si l'on utilise la matrice de corrélation, on parle alors de la transformée de Karhunen-Loève (TKL). Par la projection par ACP, les axes de projections sont orthogonaux (et décorrélés dans le cas particulier de la TKL). Nous proposons dans le chapitre 2 d'autre méthodes de réduction de dimensionnalité mais nous avons retenu dans ce travail l'utilisation de la TKL pour définir un sous-espace de dimension trois.

Ce choix de l'ACP se justifie également par notre méthode d'attribution des couleurs aux données de l'échantillon. Comme nous allons le voir dans la section qui suit, nous utilisons également l'ACP (dans sa version appelée TKL) pour calculer ces couleurs.

3.3.2 Calcul de la couleur d'une donnée de l'échantillon

Il existe de nombreux modèles de représentation des couleurs [200][146] (appelés espaces de couleurs). On peut citer par exemple :

- le modèle RVB (Rouge, Vert, Bleu, en anglais RGB, Red, Green, Blue).
- le modèle TSL (Teinte, Saturation, Luminance, en anglais HSL, Hue, Saturation, Luminance).
- le modèle CMY(K) (Cyan, Magenta, Yellow (Black))
- le modèle CIELab (défini par la Commission Internationale de l'Eclairage en 1976)
- le modèle YUV surtout utilisé en vidéo
- le modèle YIQ proche du modèle YUV

Nous avons choisi d'utiliser le modèle le plus répandu, à savoir l'espace RVB. La principale raison est la méthode d'attribution de la couleur dans notre méthode. Comme nous allons le voir, notre approche d'affectation des couleurs est purement "statistique" et ne fait pas intervenir de considérations sur la perception des couleurs [183]. La création des couleurs est complètement guidée par les données, elle n'est pas supervisée. Nous ne faisons aucun a priori sur les données et le processus ne fait pas d'apprentissage explicite.
Le codage RVB consiste à représenter l'espace des couleurs à partir de trois rayonnements monochromatiques des couleurs *Rouge*, *Vert* et *Bleu*. Nous définissons donc dans ce travail une couleur comme un triplet de valeurs : Rouge, Vert et Bleu [15]. Comme classiquement en imagerie, ces trois composantes sont codées sur un octet (un entier non signé entre 0 et 255) ce qui correspond à 256 intensités de rouge (28), 256 intensités de vert et 256 intensités de bleu, soient 16777216 possibilités théoriques de couleurs différentes (l'oeil humain ne peut en distinguer que beaucoup moins).

Dans cette partie, nous allons présenter notre solution pour associer une couleur à chaque donnée d'un échantillon. Les données ayant été projetées dans un espace de dimension trois, chaque donnée a un représentant (X, Y, Z) dans cet espace. Nous décrivons comment nous associons une couleur (R, V, B) à chaque représentant (X, Y, Z) d'une donnée. Une correspondance naïve du type X pour R, Y pour V et Z pour B peut conduire à des interprétations erronées des couleurs présentées. Aussi nous expliquons d'abord pourquoi ces approches naïves sont à éviter puis nous proposons une approche objective non triviale du calcul de la couleur d'une donnée sans aucun a priori sur la

palette des couleurs qui sera utilisée.

Nous procédons tout d'abord à un rééchantillonnage de X, Y et Z pour obtenir trois valeurs entre 0 et 255. Cette normalisation est obtenue par :

$$\begin{cases} X' &= 255 \times \frac{X - X_{min}}{X_{max} - X_{min}} \\ Y' &= 255 \times \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \\ Z' &= 255 \times \frac{Z - Y_{min}}{Z_{max} - Z_{min}} \end{cases}$$

où X_{min}, Y_{min} et Z_{min} sont respectivement les valeurs minimales de X, Y et Z sur l'échantillon de données et X_{max}, Y_{max} et Z_{max} sont respectivement les valeurs maximales de X, Y et Z sur ce même échantillon.

Nous pourrions à cette étape considérer que X', Y' et Z' représentent R, V et B à une permutation près. Malheureusement une approche de ce type n'est pas satisfaisante, les couleurs obtenues ne permettent pas de percevoir les structures ou les classes de données présentes sur l'échantillon. Même l'ordre dans lequel sont considérés X', Y' et Z' n'est pas satisfaisant, aucun argument ne permet de préférer (X', Y', Z') à (Y', Z', X').
Prenons un exemple pour éclairer ce point.
Les triplets $(255, 0, 0)$ et $(255, 255, 0)$ seront affichés et perçus comme respectivement du rouge et du jaune, alors que les triplets $(0, 0, 255)$ et $(0, 255, 255)$ seront affichés et perçus comme respectivement du bleu et du cyan. En considérant une métrique comme la distance euclidienne sur ces données projetées, on constate que les distances d'une part entre le rouge et le jaune et d'autre part entre le bleu et le cyan sont égales. En revanche la perception humaine de ces couleurs n'est pas la même. Le bleu et le cyan sont en effet plus proches entre eux que le rouge et le jaune. Ce type de représentation colorimétrique conduit donc à des rapprochements de données qui ne seront pas justifiés mais uniquement liés à une méthode de visualisation et au choix des couleurs.

Les triplets (R, G, B) et (G, R, B) ne représentant pas les mêmes couleurs, l'ordre des composantes est significatif vis-à-vis de la perception que l'on en aura. Hélas les valeurs normalisées Z', Y' et Z' n'ont aucun ordre a priori et, sans connaissance complémentaire, ces coordonnées ont le même poids au sens statistique. Pour ordonner ces trois variables, nous proposons une approche statistique en calculant les trois premières composantes C_1, C_2 et C_3 obtenues par la TKL appliquée à l'ensemble des données (voir partie *réduction de dimensionnalité*). Ces composantes sont ordonnées, la première étant plus informative au sens statistique que la deuxième et celle-ci plus informative que la troisième. Le triplet (C_1, C_2, C_3) donne des informations statistiques qui ne peuvent pas être considérées directement comme des informations colorimétriques.

La nouveauté de notre approche a été d'utiliser une transformation originale pour obtenir les composantes R, V et B. Cette nouveauté a été validée par plusieurs communications et publications [16][15][19]. Nous avons décidé de choisir les couleurs en utilisant un résultat important et connu en analyse d'images couleur. Otha, Sakai et Kanade [151] ont proposé une transformation linéaire de l'espace (R, V, B) qui simule la transformation de Karhunen-Loève pour les pixels d'une image couleur. A partir de (R, V, B), ils proposent une approximation des trois composantes de la TKL : (C_1, C_2, C_3).
Nous sommes dans la situation inverse où nous disposons des composantes (C_1, C_2, C_3) obtenues par TKL sur un échantillon de données. En considérant la transformation inverse de celle de Ohta et al., nous obtenons les triplets (R, V, B) qui approximent *Rouge*, *Vert* et *Bleu* de l'image couleur virtuelle qui aurait conduit à (C_1, C_2, C_3). Ces triplets sont définis par :

$$\begin{cases} R &= (6C_1 + 3C_2 - 2C_3)/6 \\ V &= (3C_1 + 2C_3)/3 \\ B &= (6C_1 - 3C_2 - 2C_3)/6 \end{cases}$$

A partir d'un échantillon de données décrites dans un espace de dimension trois par les triplets (X, Y, Z), nous obtenons après normalisation, TKL et transformation inverse de celle de Otha, les triplets (R, V, B) que nous proposons comme définition des couleurs des données relativement à l'échantillon considéré.

3.3.3 Construction d'une image

Dans le cas particulier où l'échantillon de données est constitué de l'ensemble des pixels d'une image multicomposante, on obtient immédiatement une image couleur à la fin de l'étape précédente. Chaque pixel multidimensionnel est représenté par un pixel couleur de composantes (R, V, B) dans l'image résultat. Dans le cas général, des données multidimensionnelles quelconques ne sont pas les pixels d'une image multicomposante et ne possèdent donc pas d'information de localisation dans l'espace image. Autrement dit, les données obtenues à l'étape précédente ne possèdent pas, a priori, de position dans l'image couleur finale. Cette partie de l'article propose une construction d'image pour visualiser ces données non initialement spatialisées.

Pour construire une image de l'échantillon de données, on associe un pixel de l'image à chaque donnée (voir *pixel-oriented visualization* [121]). Un ensemble de N données sera donc représenté par une image de N pixels. Cette approche de la visualisation orientée-pixel permet de représenter des échantillons de grande taille. Par exemple, une image 1000×1000 permet de représenter un million de données. On suppose que chaque donnée est représentée soit par un niveau de gris (un entier non signé entre 0 et 255) soit par une couleur (un triplet (R, V, B)) qui lui est associé. Les pixels (donc les représentations

des données) doivent être placés spatialement dans l'image, ce paragraphe de l'article présente comment les coordonnées de chacun des pixels sont déterminées.

Les pixels ne peuvent pas être placés arbitrairement dans l'image. En effet, ceci produirait une image où les données seraient dispersées et l'oeil humain ne pourrait que très difficilement identifier des groupes ou classes de données. Les similarités ou dissimilarités entre données seraient alors difficiles à déterminer, les rapprochements entre données dispersées étant difficiles à effectuer. Pour que l'image soit lisible au premier coup d'oeil de manière très intuitive, il faut que les données similaires soient spatialement très proches, les données dissimilaires éloignées et que les classes de données soient le plus connexes possible. Cette information spatiale contenue dans l'image sera redondante avec l'information colorimétrique que nous avons proposée précédemment. En effet, plus les données seront similaires et plus leurs couleurs seront proches mais aussi plus leurs localisations dans l'image seront proches. Ce n'est qu'à cette condition de redondance que la lecture de l'image sera intuitive et immédiate. Nous proposons une construction de ce type d'image avec deux étapes : les pixels (i.e. les couleurs associées aux données) seront d'abord rangés de manière à former une ligne de pixels successifs, ensuite cette ligne sera utilisée pour remplir l'image.

Ranger les pixels (c'est-à-dire trouver un ordre pour l'ensemble des pixels) équivaut à projeter les données sur un espace de dimension un. Les données sont alors ordonnées ou rangées par l'ordre naturel sur l'espace $\mathbb{R}$ (espace de projection de dimension un). Il serait possible de refaire une étude sur la meilleure projection sur un espace 1D mais nous avons déjà proposé une projection 3D par la TKL. Avec ce type de transformation, la meilleure projection 1D sera obtenue par la première composante principale. Cependant, pour tenir compte des trois premières composantes principales et non uniquement de la première, nous rangeons les données par un tri avec trois clefs successives qui sont les trois premières composantes principales de la TKL. Les représentations de ces données (i.e. les pixels) se trouvent ainsi rangés formant une ligne de pixels successifs.

Dans la deuxième étape de la construction de l'image représentative de l'ensemble des données, il faut remplir l'image par cette ligne de pixels successifs. Les courbes de Peano constituent le moyen le plus classique pour effectuer cette construction [147], les courbes en "U" de Hilbert ou les courbes en "Z" de Morton sont les plus utilisées. Le principal avantage de ces courbes est de préserver au mieux la connexité des classes de données [168]. La procédure de construction de telles courbes est récursive, nous avons choisi d'utiliser les courbes en "U" de Hilbert dont on peut voir les premières étapes de la construction récursive sur la figure 3.13. La raison qui nous a fait préférer la courbe en "U" à celle en "Z" est que les sauts y sont moins brutaux. Il y a ainsi moins de risque de

"découper" une zone correspondant à une classe de données. Il y a donc plus de chance de voir apparaître ces classes comme des zones connexes sur l'image résultat.

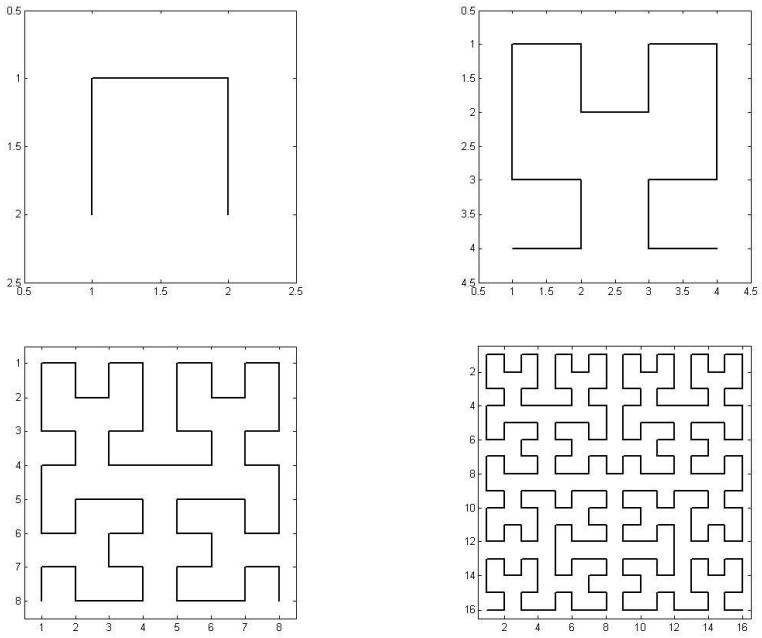


FIG. 3.13 – Etapes de la construction récursive d'une courbe de Hilbert (dite en "U").

Les deux étapes de rangement des données puis de parcours de l'image par une courbe d'Hilbert évitent de disperser les pixels dans l'image construite. Cette approche tend à préserver la cohérence spatiale des données dans leur représentation par une image permettant ainsi une visualisation très intuitive des ensembles de données.

3.3.4 Applications

Nous proposons dans cette partie plusieurs applications de notre méthode. Nous allons tout d'abord l'utiliser sur des données simulées, en conditions contrôlées, afin de pouvoir valider l'efficacité de chacune des étape qui la composent (section 3.3.4.1). Ensuite elle a été appliquée à des images multicomposantes "réelles" (section 3.3.4.2) et à deux bases de données "réelles" IRIS (section 3.3.4.3), et "ForestCoverType", disponible également sur [150].

3.3.4.1 Visualisation des classes sur des données simulées

Pour valider notre méthode et vérifier son efficacité, nous l'avons utilisée avec des données choisies :

- sur une image en "vraies" couleurs, afin d'obtenir une évaluation qualitative de la visualisation
- sur une image multicomposante simulée, pour valider notre étape de *color mapping*
- et sur des données multidimensionnelles simulées, pour valider l'étape d'organisation spatiale des pixels

Nous avons donc tout d'abord appliqué notre technique de visualisation à une image en vraies couleurs. Nous allons vérifier à l'aide de cet exemple, que notre approche purement "statistique" d'attribution des couleurs est cohérente.
Comme nous l'avons déjà souligné, une image couleur peut être considérée comme multicomposante. Les composantes correspondent aux images en niveaux de gris obtenues partir du canal rouge, du vert et du bleu. Il pourrait sembler incongru de vouloir utiliser une méthode de visualisation pour une image qui est déjà visualisable naturellement. Mais l'objectif de cette application est justement de pouvoir comparer le résultat de notre technique avec la visualisation instantanée. Notre méthode fournit une image en fausse couleur qu'il est intéressant de comparer à l'image originale. Cependant il faut rappeler que le but de notre technique est de visualiser et séparer visuellement les classes, pas de reconstituer les couleurs (ce qu'elle ne fait d'ailleurs pas comme nous allons le voir). Notons que dans cette application, la réduction de dimensionnalité et le remplissage ne sont pas utilisés, ce qui permet de cibler la validation sur l'attribution des couleurs. Les données possèdent déjà 3 attributs, et une localisation dans l'image finale.

L'image initiale -donc l'ensemble de données à visualiser- est représentée sur la figure 3.14 et l'image générée par notre technique sur la figure 3.15.

Quel que soit le modèle de couleurs est toujours possible de décomposer une couleur en deux parties. La partie achromatique qui correspond à la quantité de gris qu'elle contient et la partie chromatique qui détermine la teinte de la couleur [200]. Dans la lecture d'une image, les informations achromatiques ont plus d'importance que les chromatiques. En effet une image en niveaux de gris, sans information chromatique, est interprétable par le cerveau humain. Les formes d'une scène peuvent être compréhensibles directement à l'aide d'une image en niveaux de gris. On peut donc dire qu'une bonne attribution de couleurs doit conserver ces informations contenues dans une image couleur.
Les figures 3.16 et 3.17 présentent les images de la luminance de l'image initiale et la luminance de l'image générée par notre méthode. Nous avons calculé la luminance en effectuant une moyenne pour chaque pixel des valeurs de ses composantes rouge, verte et bleue. Le coefficient de corrélation entre les deux luminances est égal à 1. Cette

FIG. 3.14 – Image en "vraies" couleurs constituant l'ensemble de données à visualiser

FIG. 3.15 – Image en "fausses" couleurs, fournie par notre méthode de visualisation

FIG. 3.16 – Luminance de l'image initiale de la figure 3.14

FIG. 3.17 – Luminance de l'image générée par notre technique (Figure 3.15)

application de notre méthode à une image de 500×375 pixels montre donc que notre *color mapping* conserve la principale information achromatique contenue dans une image couleur.
L'information de chrominance étant la plus importante pour la perception humaine, ce résultat valide donc l'efficacité visuelle de notre étape d'affectation des couleurs. On peut aussi constater que le contraste entre les objets (la tasse et le citron par exemple) est conservé ainsi que toutes les proximités entre les couleurs (le citron et l'orange par exemple).
Cependant il faut noter que l'information chromatique, quant à elle, n'est pas conservée par notre technique.

Nous avons ensuite appliqué notre méthode en conditions contrôlées. Pour cela, des échantillons de données ont été simulés. Ces données synthétiques permettent de vérifier l'efficacité de la méthode de visualisation.

Dans un premier exemple (voir figure 3.18), nous avons généré une image multicomposante de 256×256 pixels en dimension 6. Chaque composante est une composante binaire : la valeur d'un pixel sur une composante ne peut prendre que deux valeurs possibles de niveaux de gris. Nous avons simulé cet ensemble de 65.536 données de manière à former 12 classes de pixels. Cependant une telle image aurait pu être, en théorie composée de 2^5 classes.
Sur chacune des composantes, les pixels sont distribués de manière très lisible, puisque regroupés en régions par valeurs. Pourtant, visuellement, la segmentation de cette image n'est pas simple. La segmentation nécessite donc l'utilisation d'un processus de fusion pour pouvoir être révélée.

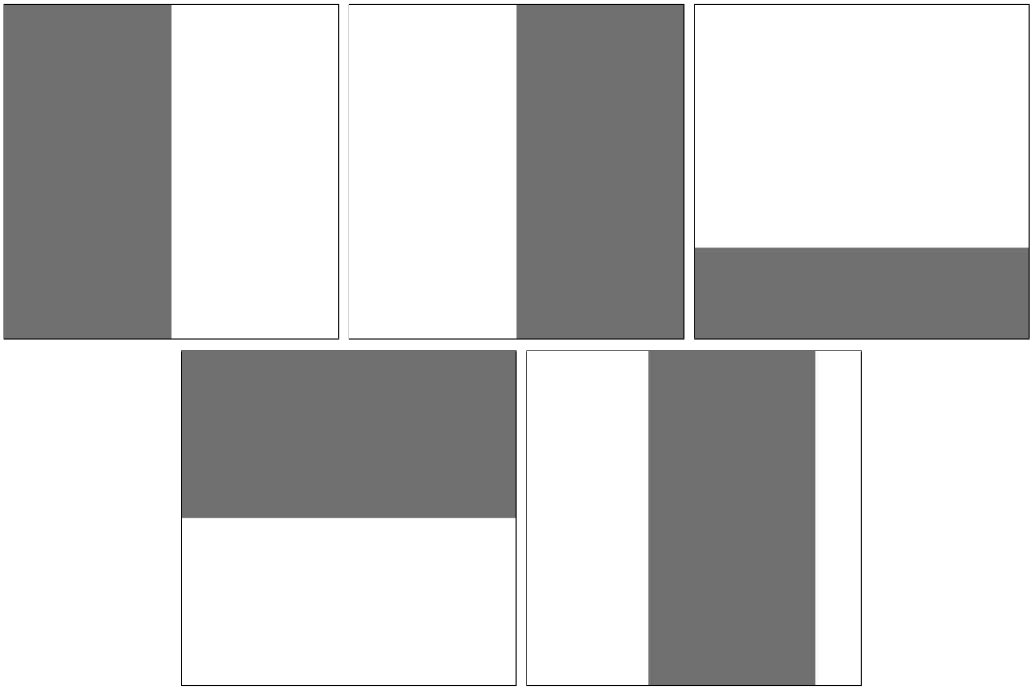


FIG. 3.18 – Les six composantes d'un échantillon de 65.536 données (image 256×256 à six composantes)

L'image résultat générée par notre méthode est présentée sur la figure 3.19. On peut observer distinctement les classes de pixels. Chaque classe de pixel correspond à une couleur unique. Les classes sont ainsi visibles directement et la segmentation est immédiate. Par ailleurs, les couleurs différentes sont informatives sur la proximité des classes. Cette information supplémentaire n'était pas du tout visible en regardant les composantes principales obtenues à la première étape de notre processus. Cet exemple démontre l'intérêt de notre étape d'affectation des couleurs aux données. En agissant comme un processus de fusion d'information, elle permet de révéler les structures cachées inhérentes aux images multicomposantes. L'application à des données réelles le confirmera par la suite.

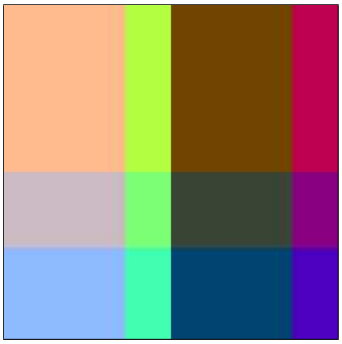


FIG. 3.19 – Visualisation couleur de 65.536 données d'un espace de dimension six : image couleur 256×256 permettant de visualiser les 12 classes de données.

Nous allons maintenant, sur un second exemple, montrer l'utilisation de la technique de remplissage qui nous permet de construire une image à partir de données non spatialisées.

Nous avons cette fois simulé un échantillon de 65.536 données multidimensionnelles en dimension 6 sans leur attribuer d'organisation spatiale. Ces données ne sont donc cette fois pas des images. Chaque attribut est à valeur binaire, comme dans l'exemple précédent. Pour mieux se représenter ces données nous les présentons sous forme d'images de 512×128 pixels sur la figure 3.20. Nous avons cette fois simulé 16 classes de données. Nous avons calculé deux images résultats : la première, présentée sur la figure 3.22, est le résultat après l'affectation des couleurs, et en utilisant le même affichage que pour les composantes ; la seconde (figure 3.23) est le résultat produit à la fin du processus en entier, c'est à dire après avoir organisé spatialement les données en utilisant le parcours de Peano. Sur la première figure, malgré l'affectation des couleurs, il est impossible de distinguer les classes. L'utilité de l'arrangement des données est ici spectaculaire. En effet la visilité des classes sur cette seconde figure est instantanée. Cet exemple plaide donc en

faveur du parcours et montre également qu'il peut être intéressant d'utiliser la méthode dans son intégralité sur des images et pas seulement avec des données qui n'en sont pas.

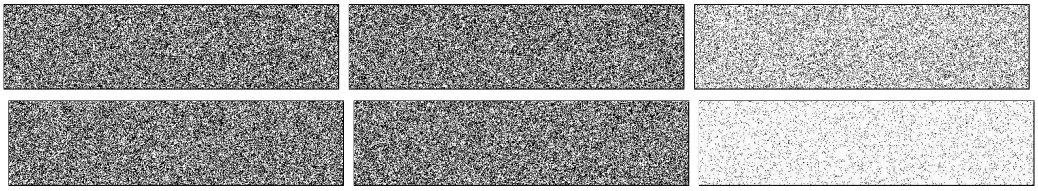


FIG. 3.20 – Les six composantes d'un échantillon de 65.536 données (image 256×256 à six composantes)

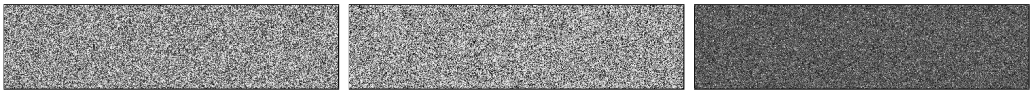


FIG. 3.21 – Les trois premières composantes principales de l'échantillon de 65.536 données.

FIG. 3.22 – Visualisation couleur de 65.536 données d'un espace de dimension six : image couleur 256×256 permettant de visualiser les 16 classes des données, non spatialement organisées. (image grossie)

Une bonne méthode de visualisation devrait permettre d'identifier au premier coup d'oeil les différentes classes présentes parmi les données pour rendre compte des similarités entre ces données. Sur ces deux exemples il n'y a que 12 et 16 classes simulées. Ces classes sont directement visualisées sur l'image couleur résultat. Ces deux exemples très simples montrent l'efficacité de la méthode de visualisation pour présenter les différentes classes de données (i.e. les similarités de données) par des couleurs. Cette visualisation présente l'avantage d'être immédiatement lisible par l'oeil humain et facile à interpréter. Notre système de visualisation permet de classer les données et de leur affecter un label sous forme d'une couleur associée à chacune des classes de l'échantillon. Ces couleurs indiquant une similarité entre données, elles donnent aussi une indication sur la proximité des différentes classes. Dans cette partie, nous avons pris un exemple très simple

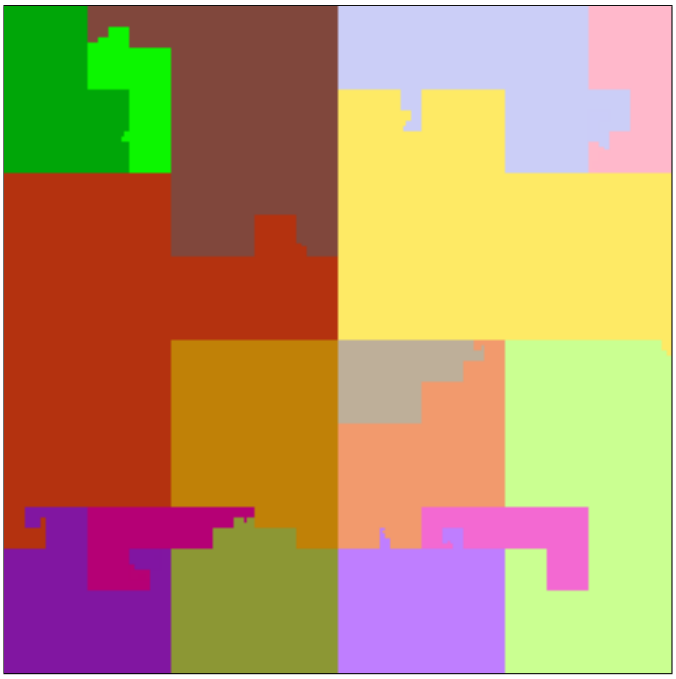


FIG. 3.23 – Visualisation couleur de 65.536 données d'un espace de dimension six : image couleur 256×256 permettant de visualiser les 16 classes de données, spatialisées (image grossie).

pour montrer l'efficacité de la méthode, des exemples plus complexes avec des données simulées donnent des résultats similaires. Cet outil de visualisation se révèle donc bien adapté aux grands échantillons de données multidimensionnelles ; il permet de visualiser les principales structures de l'échantillon lorsque la réduction de dimension à trois conserve l'information sur ces structures.

3.3.4.2 Visualisation d'images multicomposantes

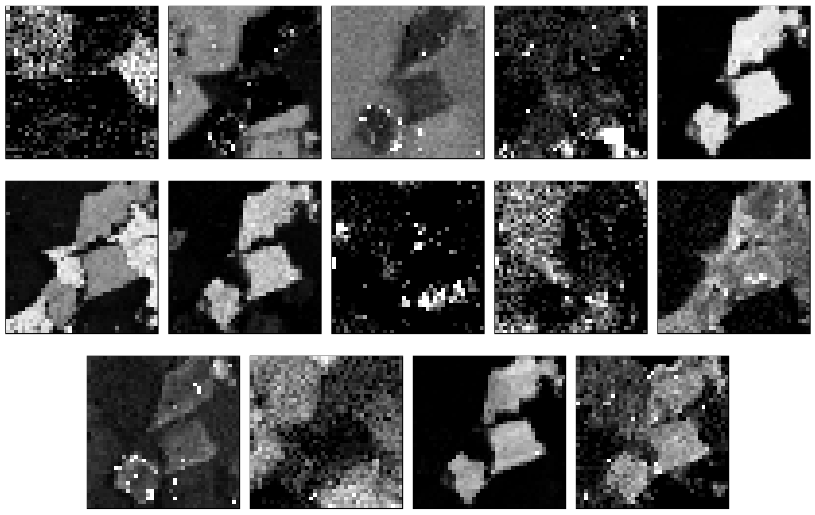
Dans le cas réel d'images multicomposantes, nous observons des résultats semblables : les différentes classes de données (dans ce cas des ensembles de pixels) sont le plus souvent immédiatement visualisées par notre méthode avec une simple image couleur. Pour illustrer ce propos nous considérons un ensemble de 14 cartes de concentration d'éléments chimiques d'un spécimen de granite enregistrées en fluorescence X [197]. Cet ensemble constitue une image de 14 composantes et de 39×40 pixels. Nous pouvons appliquer notre méthode de visualisation aux 1560 données (ou pixels) appartenant à un espace de dimension 14 (voir Fig.3.24).

L'image couleur que nous proposons permet de donner un résumé des 14 composantes de cette image de fluorescence X. Différentes méthodes de classification ont permis d'établir que quatre phases sont présentes sur cet exemple d'image multicomposante ([197], [94]) ; ces quatre phases sont nettement observables sur l'image couleur que nous proposons pour visualiser l'image multicomposante. Cet exemple réel et non synthétisé confirme que notre méthode de visualisation par les couleurs permet bien de révéler les différentes classes de données multidimensionnelles. Notons à nouveau que dans ce cas particulier des images multicomposantes, il est inutile d'utiliser l'étape de construction de l'image utilisant le parcours de Peano.

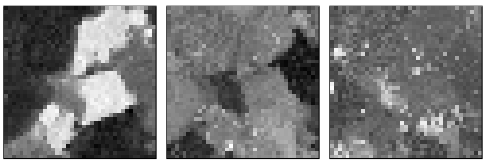
Après avoir appliqué notre méthode sur des données simulées et vérifié la validité de notre méthode, nous l'avons utilisée sur des bases de données classiques et dont nous connaissons toutes les caractéristiques : "IRIS" et "ForestCoverType" [150].

3.3.4.3 Visualisation de bases de données multidimensionnelles réelles

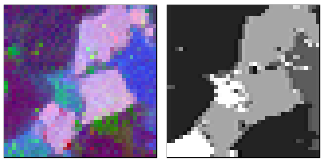
Nous appliquons notre méthode de visualisation à la classique base de données IRIS de Fisher [12]. Cet ensemble de 150 données en dimension 4 (longueur des sépales, largeur des sépales, longueur des pétales, largeur des pétales) est composé de trois classes : iris setosa, iris versicolor et iris virginica, avec 50 observations ou données par classe. A partir de ces 150 données en dimension 4 sont pouvons calculer une image couleur 16×16 (voir Fig.3.25) que l'on peut comparer avec l'image des labels des trois classes.



Les 14 composantes initiales



Les 3 premières composantes principales



L'image couleur obtenue et carte des 4 phases

FIG. 3.24 – Image à 14 composantes visualisée par une seule image couleur et les quatre phases détectées par une méthode de classification

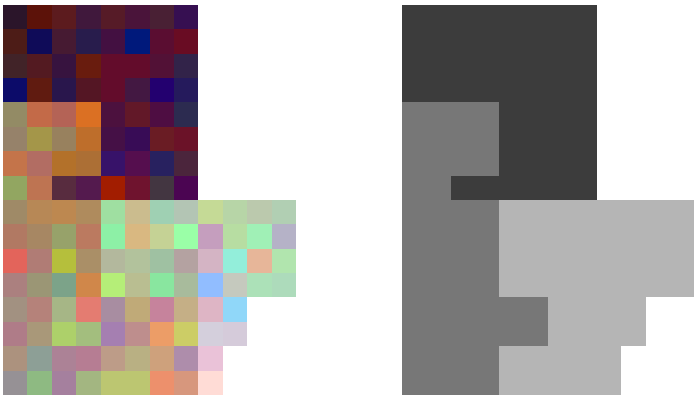


FIG. 3.25 – Visualisation des 150 données IRIS de dimension 4 (image couleur et label des trois classes)

La classe des iris setosa est facilement différentiable des deux autres classes, en revanche ces deux classes sont difficiles à séparer. Comme nous l'avons déjà précisé au chapitre 2, ce résultat est bien connu sur cette base de données. Nous le retrouvons avec notre méthode de visualisation bien que celle-ci soit peu adaptée à un échantillon de faible dimension (dimension quatre) et de faible effectif (150 données).

Voici maintenant un second exemple d'application à des données "réelles". Nous considérons ici la base de données "Forest Cover Type". Cette base de données est un ensemble d'observations de parcelles de 30×30 mètres relevées par le *US Forest Service*. La base initiale était constituée de 581012 observations, et de 54 attributs. Nous avons extrait un sous-échantillon de cette base, constitué de 1024 observations (pour des raisons d'affichage), en ne conservant que les 10 variables quantitatives. Cet échantillon contient 7 "vraies" classes. Nous disposons donc d'un ensemble de données :

- contenant 1024 points
- en dimension 10
- composé de 7 classes

Le résultat fourni par notre méthode est assez bon et permet de distinguer un découpage, et donc une structure (même grossière) de l'ensemble des données (voir la figure 3.26).

Remarque : Le résultat n'est pas aussi démonstratif que pour les données IRIS. Ceci est dû au fait que nous éliminons les variables non quantitatives et nous privons ainsi de l'information apportée par ces 44 variables. Par ailleurs, on peut supposer que la réduction de dimensionnalité, faisant passer d'un espace de dimension 10 à un espace de dimension 3, laisse échapper un peu d'information supplémentaire. Toutefois, et compte tenu de ces

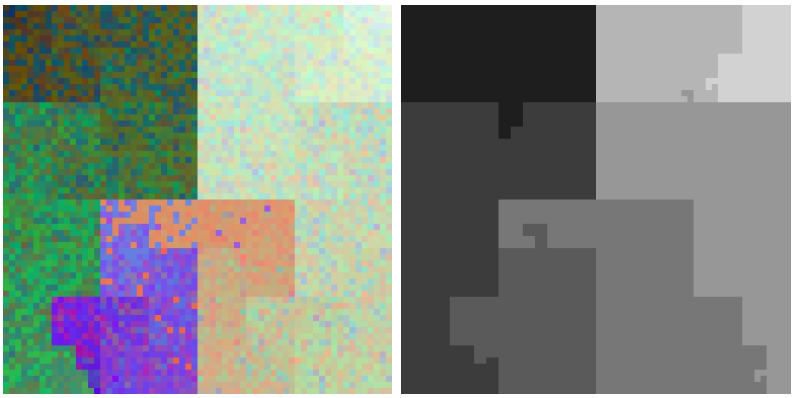


FIG. 3.26 – Visualisation des 1024 données extraite de la base “Forest Cover Type”, de dimension 10 (image couleur et label des quatre classes)

contraintes, le résultat est tout à fait satisfaisant pour une visualisation "exploratoire" préliminaire à une analyse plus fine. Ce constat confirme l'intérêt que peut présenter notre méthode dans ce type d'utilisation.

3.4 Visualisation dynamique pour l'exploration de données multidimensionnelles

Avec notre méthode, nous disposons d'un outil efficace de visualisation statique des données [15][18][20]. Nous travaillons actuellement sur le développement d'un outil de visualisation dynamique pour l'exploration de masses de données utilisant notre technique statique. Nous présentons dans cette partie le concept sous-jacent, ainsi qu'un premier exemple d'utilisation.

3.4.1 Concept

Nous travaillons donc actuellement sur le développement d'une méthode basée sur notre principe de visualisation. L'objectif de cet outil est l'exploration dynamique d'un ensemble de données.

Le principe repose sur une modification de la Transformée de Karhunen-Loeve. En effet la théorie de la TKL (ou de l'ACP) repose tout d'abord sur un centrage des données. Cette transformation est liée au calcul de l'inertie d'un nuage de point. En effet l'inertie, comme on peut le voir dans le chapitre préliminaire de ce mémoire, est une quantité calculée par rapport au barycentre, ou point moyen, du nuage de point considéré. La

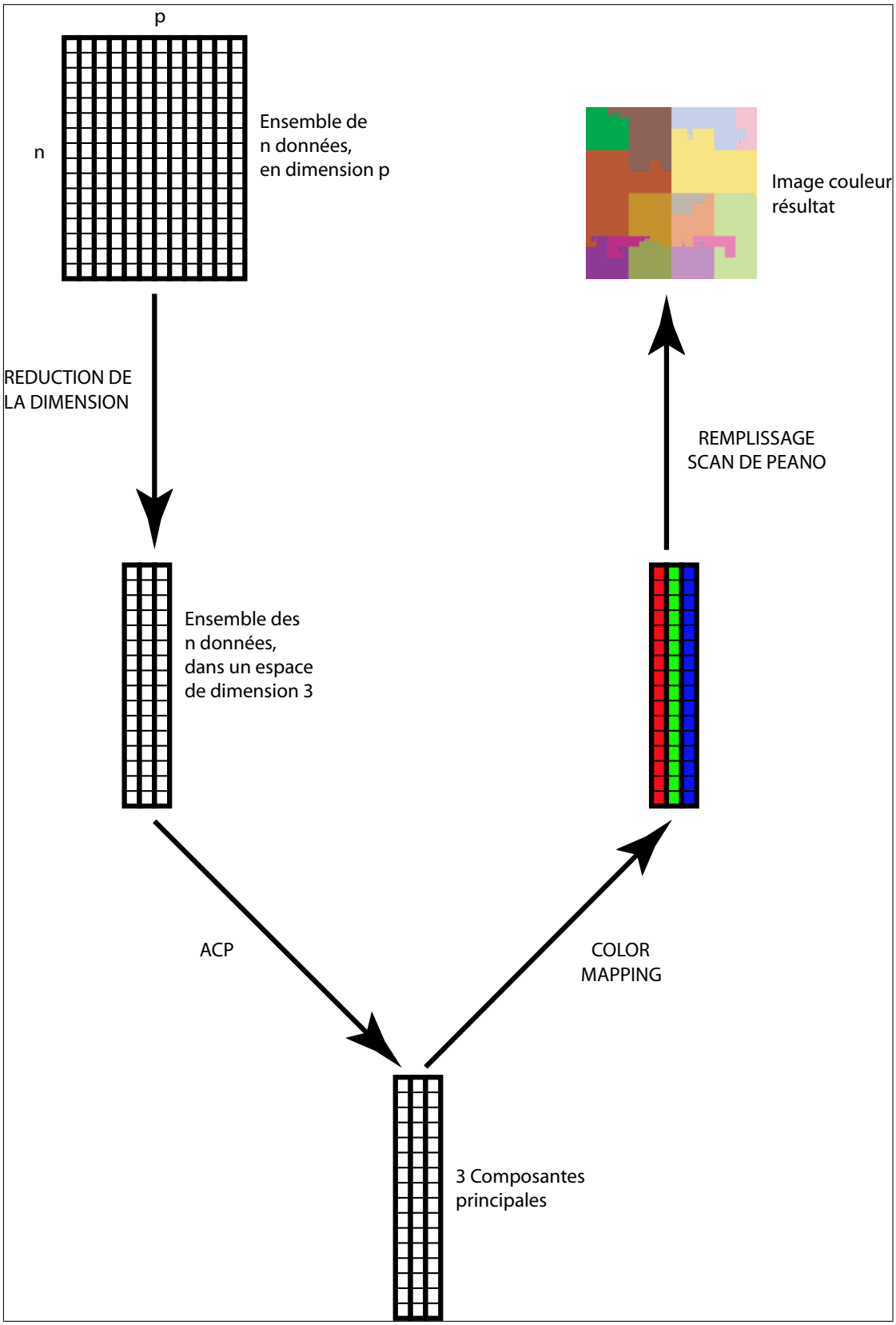


FIG. 3.27 – Schéma représentant le processus de création de l'image couleur à partir des données initiales

première idée de ce nouveau développement est donc de changer le point de référence pour le calcul de l'inertie. La quantité calculée peut alors être vue comme une *inertie relative*. Une autre façon de voir les chose serait de considérer un nouveau nuage de point constitut du nuage initial auquel on aurait ajouté un nouveau point -le point de référence considéré- pondéré suffisamment pour lui accorder une grande importance dans le calcul. Le reste du processus se déroule alors comme dans la méthode initiale.
Cette modification conduit alors, par application de notre technique, à obtenir une image différente selon le point de référence considéré. Le point de référence peut être défini comme un observateur des données. L'observation correspondant alors à l'image fournie en considérant l'observateur comme point de référence.
Avant de procéder à une exploration dynamique, une première étape d'initialisation est nécessaire. Cette initialisation consiste à fixer la spatialisation des pixels associés aux données. Autrement dit, il faut déterminer une fois pour toute une position dans l'image pour chacune des données. Cette initialisation permet ainsi, quelle que soit l'image, de pouvoir identifier une donnée à une même position sur l'image .

La seconde idée est donc de générer et d'observer les images en déplaçant l'observateur dans l'espace des données à explorer. On obtient alors une séquence d'images lors de l'exploration par déplacement de l'observateur. On associe une image couleur à chaque position de l'observateur. Ce déplacement visuel au sein de l'ensemble de données suggère plusieurs questions :

- Comment définir la trajectoire de l'observateur ?
- A quelle fréquence doit-on générer (photographier ?) les images ?

Trajectoire. Pour la première question, plusieurs solutions s'offrent à nous. Dans le cadre de ce travail (voir partie suivante d'exemple d'application), nous n'avons jusqu'à présent mis en place que deux types de déplacements : un déplacement "basique" qui définit une trajectoire rectiligne arbitraire à travers l'espace des données, et un déplacement aléatoire "contrôlé" (les déplacements sont aléatoires mais une direction est privilégiée). Comme nous le verrons alors, ce choix de déplacement -pourtant rudimentaire- conduit à des résultats très intéressants. Ce constat n'apporte que plus d'intérêt à l'utilisation de trajectoires plus fines. On peut distinguer plusieurs types de constructions de trajectoires :

- des trajectoires de type "marches aléatoires"
- des trajectoires guidées par les données (*data driven*), qui conduirait l'observateur à aller vers des zones intéressantes de l'ensemble des données de façon automatique
- des trajectoires guidées par des requêtes (*query dependent*) préalables de l'utilisateur, qui suggèrent une connaissance a priori et/ou une intention dans l'exploration
- des trajectoires interactives où l'observateur est guidé par l'utilisateur qui interprète

en temps réel la séquence d'images construites

Vitesse. La fréquence des images est un problème très lié au précédent. On peut en effet imaginer les mêmes types d'approche : une fréquence dépendante des données, des requêtes ou interactive. La fréquence de génération des images au cours du déplacement de l'observateur induit implicitement une notion de vitesse de déplacement. En effet, modifier la fréquence de "photographie" sur la trajectoire revient à considérer une modification de la vitesse d'un observateur pour lequel les images seraient générées à intervalles réguliers. Autrement dit : augmenter la fréquence de création des images sur une partie de la trajectoire équivaut à diminuer la vitesse de déplacement, sur cette partie, d'un observateur "photographiant" à intervalle fixe. Ce paramètre présente un grand intérêt en permettant par exemple de visualiser avec plus d'attention -plus de temps- des zones présentant plus d'intérêt de l'ensemble de données, et à l'inverse, de passer rapidement dans les endroits moins denses ou vides. Ce type d'approche permet alors de ne pas noyer les informations pertinentes en les représentant par une trop courte séquence d'images. Il est en effet important de pouvoir s'attarder autour de points particuliers afin d'étudier la structure des données dans des endroits cruciaux comme les frontières entre les classes, les centres des classes ou d'autres points particuliers.

3.4.2 Application à la base de données IRIS

Considérons par exemple l'exploration dans la base de données IRIS. On décide, dans cet exemple de définir une trajectoire aléatoire (mais contrôlée à l'aide de paramètres pour privilégier une direction) à travers l'ensemble de données (voir figure 3.28). 1000 images sont générées à intervalles réguliers. En observant la séquence d'images on constate des changements de couleurs et des modifications de la structure visible. On peut voir ce changement sur les images de la figure 3.29. L'utilisation de plusieurs images différentes permet ainsi de “séparer” visuellement les classes (on rappelle que la base de données IRIS est constituée de 3 classes) comme le montre le schéma de la figure 3.30.
Les positions de l'observateur lors de ces changements sont des points intéressants de l'espace. Ces points singuliers, fournissant des ”points de vue différents” des données semblent jouer un rôle important. Dans des développements futurs, nous envisageons d'étudier plus précisément le rôle que jouent ces points.
L'exploration dynamique nous a permis de reconstituer les classes constituant les données, visuellement. Les séparations entre les classes sont déterminables rapidement. Cet outil se révèle donc très intéressant pour la fouille de données et pourrait jouer des rôles différents, en confirmant des avis d'experts ou en attirant l'attention de ces experts sur des points particuliers.

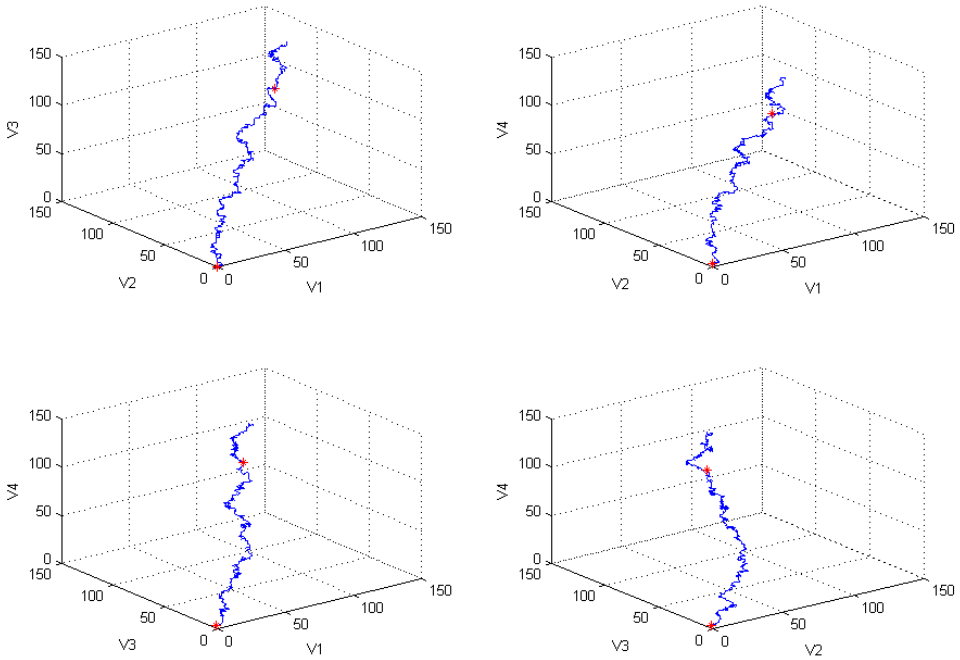


FIG. 3.28 – Définition d'un parcours aléatoire à travers la base de données IRIS pour l'exploration dynamique. Chaque graphique représente la trajectoire dans l'espace de projection défini par les variables prises trois par trois. Les points rouges sur la droite représentent les positions de l'observateur qui ont généré les images de la figure 3.29

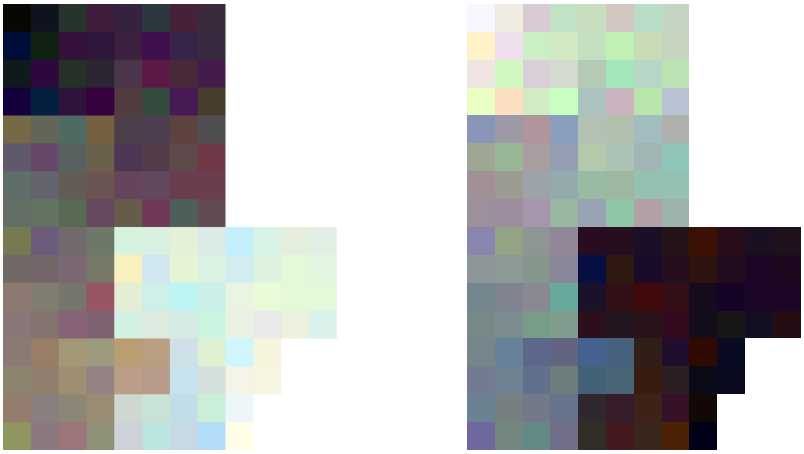


FIG. 3.29 – Images extraites de la séquence obtenue lors du parcours défini à la figure 3.28. Ces deux images se succèdent en passant le point défini par la croix rouge de ladite figure

La méthode d'exploration de données que nous venons d'exposer semble pouvoir être un outil riche pour la découverte de structures en fouille de données. Elle fera l'objet de travaux et d'études ultérieurs pour être validée. Les deux caractéristiques de déplacement, à savoir *trajectoire* et *variations de la vitesse de déplacement* doivent notamment être étudiées plus précisément et faire l'objet d'expérimentations poussées. Les aspects :

- guidé par les données
- guidé par requêtes
- interactivité

peuvent également faire l'objet d'études futures. Enfin le rôle des points singuliers peut être également l'objet d'attentions particulières.

3.5 Discussion et conclusion

La méthode que nous proposons (voir schéma récapitulatif sur la figure 3.27) pour visualiser un ensemble de données multidimensionnelles permet une représentation rapide de tout l'ensemble par une simple image couleur. Avec cet outil de visualisation, la couleur est un auxiliaire fondamental qui permet une lecture directe et très intuitive de l'image et donc de l'ensemble des données. L'utilisation d'une image permet ainsi d'obtenir une méthode de visualisation particulièrement bien adaptée aux grands échantillons de données multidimensionnelles et en particulier aux images multicomposantes.

Notre technique de visualisation a ses limites. Le fait d'avoir réduit la dimension à

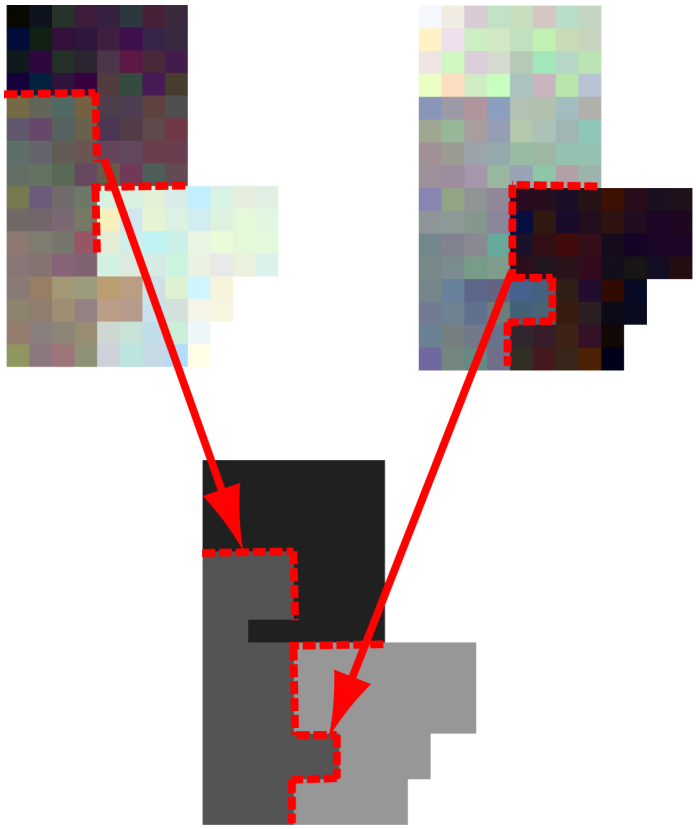


FIG. 3.30 – Reconstitution de la "vraie" classification à partir des frontières visibles sur les images de la figure 3.29. Seule une petite partie de cette frontière n'est pas vraiment visible.

trois pour calculer les couleurs est un facteur limitatif à l'efficacité de cette approche. En effet, lorsque nous travaillons sur des données de grandes dimensions, il est nécessaire de réduire la dimension sans connaître la dimension intrinsèque de nos données. Il est sûr que, si cette dimension intrinsèque est supérieure à trois, alors une partie éventuellement importante de l'information sera invisible dans notre représentation. Un autre inconvénient des travaux présentés ici est l'utilisation d'une méthode linéaire de projection pour réduire la dimension. Ce modèle linéaire de réduction de dimensionnalité n'est pas nécessairement le mieux adapté aux données à traiter. D'autres approches, par exemple par des cartes auto-organisatrices [126][127], pourraient être envisagées. Il faut cependant remarquer que les trois premières composantes d'une ACP (on peut d'ailleurs donner le pourcentage de variance expliquée dans les trois premières composantes) sont le plus souvent largement suffisantes pour que notre méthode de visualisation soit applicable à la plupart des cas.

Notre approche est très objective, l'observateur ne choisit pas les couleurs utilisées et n'apporte aucune connaissance a priori sur les données. Les couleurs dépendent uniquement de l'échantillon de données utilisé. Si l'échantillon change, les couleurs changent aussi. La couleur n'est donc interprétable que relativement aux autres couleurs de l'image (i.e. aux autres données). L'objectivité est un avantage considérable mais notre méthode est peu robuste. En effet, si l'échantillon de données contient de nombreuses données aberrantes, les couleurs des classes de données peuvent s'avérer difficiles à discriminer, l'image étant alors difficile à lire. Il faudrait alors apporter de la subjectivité pour améliorer la robustesse et la fiabilité de la méthode.

Les limitations citées précédemment ne sont pas spécifiques à cet outil de visualisation. Comme pour toutes les techniques de visualisation, il est nécessaire de confirmer les observations par d'autres moyens d'analyse et de traitement des données. Notre méthode de visualisation des données multidimensionnelles offre un outil nouveau qui peut être utilisé principalement dans deux cas : soit pour approche préliminaire à une exploration ou une classification non supervisée des données, soit pour contrôler ou confirmer le résultat d'une analyse des données obtenue par d'autres méthodes. La simplicité de cet outil de visualisation est un atout dû essentiellement à l'utilisation de la couleur. Son utilisation pour l'exploration dynamique permet la recherche et la découverte de structures dans un ensemble de données. Ces riches perspectives sont à valoriser, dans un travail à venir, par des publications et des communications.
Tous ces éléments font de notre méthode de visualisation un auxiliaire précieux dans l'exploration d'images multicomposantes et plus généralement l'exploration de grandes masses de données.

Chapitre 4

Représentation floue pour la classification automatique

J'ai adopté un nouveau système de classement du travail à effectuer : urgent, très urgent, trop tard.
Philippe Gelück (Le Chat)

Sommaire

4.1 Introduction 90
4.2 Survol des méthodes de classification 94
4.2.1 Méthodes hiérarchiques 94
4.2.2 Méthodes de partitionnement 98
4.2.2.1 Méthodes de réallocation 99
4.2.2.2 Méthodes basées sur la densité de probabilité 101
4.2.3 Méthodes de classification floue 107
4.2.4 Conclusion 107
4.3 Une nouvelle représentation floue des données d'un échantillon 108
4.3.1 Exemple introductif 109
4.3.2 Représentation floue des données 112
4.3.2.1 Passage aux rangs 114
4.3.2.2 Les données comme sous-ensembles flous 114
4.3.2.3 L'échantillon comme sous-ensemble flou 116
4.3.2.4 Ensembles flous de connexion 120
4.4 Application à la classification 121
4.4.1 Principe 121
4.4.2 Exemples et applications 124

4.4.2.1 Données simulées 124
4.4.2.2 Image multicomposante 126
4.4.2.3 Données réelles 133
4.5 Discussion et conclusion 134

4.1 Introduction

"La nature offre un grand nombre de populations qu'il est souhaitable de répartir en catégories" commencent G. Celeux et col. dans [39]. La classification est en effet utilisée dans la plupart des disciplines scientifiques. Les exemples d'utilisation sont nombreux et variés : classification des pathologies en médecine, des populations en épidémiologie, classification de gènes dans l'étude des génotypes, segmentation (classification de pixels) en imagerie, taxonomie[1] en botanique, segmentation de clientèle en marketing ou encore apprentissage non-supervisé en intelligence artificielle.
Les méthodes de classification automatique sont des méthodes de structuration. Les résultats d'une classification assurent une vue concise et structurée des données. Ils permettent de faire ressortir le pouvoir séparateur ou non des paramètres.
Lorsqu'il est amené à employer une méthode de classification, l'utilisateur dispose d'un tableau d'individus (en lignes) décrits par des variables (en colonnes), ou d'un tableau de dissimilarités (ou de distance), ou même d'un tableau de contingence. Il suppose que des regroupements existent ou bien exige que certains regroupements soient effectués. Ils sont obtenus à l'aide d'algorithme, sous forme de partitions ou de hiérarchies de partitions. Les groupes sont appelés *classes*.
Les classes sont donc des groupements de données de sorte que :

- deux données d'une même classe sont aussi semblables que possible
- deux données appartenant à deux classes différentes sont aussi dissemblables que possible

L'intérêt des classes produites peut être de confirmer l'existence réelle supposée d'une partition ou bien simplement de constituer des outils permettant une exploration des données.
Les algorithmes de classification peuvent être divisés en deux familles : les algorithmes de partitionnement et les algorithmes hiérarchiques.
Les algorithmes de partitionnement conduisent directement à une partition contrairement aux algorithmes hiérarchiques. Les algorithmes hiérarchiques *agglomératifs* construisent les classes par agglomérations successives des objets deux à deux. Les algorithmes agglomératifs produisent donc une hiérarchie de partitions. Les algorithmes descendants forment la famille antagoniste des algorithmes agglomératifs. Ces algorithmes procèdent

[1] Le terme de "taxonomie" est parfois utilisé pour "classification automatique"

par dichotomie et fournissent également des hiérarchies de partitions. Les techniques de partitionnement et hiérarchiques peuvent être utilisées conjointement et forment alors des algorithmes dits *mixtes* [206].
Nous venons de voir qu'il existait deux types de méthodes hiérarchiques. La famille des algorithmes de partitionnement peut quant à elle être divisée en plusieurs sous-catégories. On peut notamment distinguer :

- les méthodes de réallocation
- les méthodes basées sur la densité (*density-based*)
- les méthodes basées sur les graphes (*graph-based*)
- les méthodes probabilistes
- k-médoïdes

Nous passerons en revue les techniques les plus efficaces et les plus classiques de ces méthodes dans la partie 4.2.

Le leitmotiv de notre travail est le traitement des images multicomposantes. Plus précisément, un des buts de l'équipe de Traitement d'Images de notre laboratoire est de développer des outils de traitement et d'analyse d'images multicomposantes. La classification de pixels d'images multicomposantes est une des problématiques qui nous concernent donc ici. Les pixels d'images multicomposantes sont des données particulières. Les spécificités de ce type de données induisent des contraintes sur les méthodes de classification à utiliser [184] :

Grand nombre de données Le nombre de pixels, et donc de données, d'une image multicomposante est généralement très élevé ;

Forme des classes Les classes de pixels d'une image multicomposante ont souvent des formes non-convexes dans l'espace paramétrique ;

Dimensionnalité élevée l'amélioration des moyens d'acquisition n'a pas seulement pour conséquence l'augmentation du nombre de pixels mais aussi celle du nombre de composantes ;

Bruit et données aberrantes De nombreux matériels d'acquisition fournissent des images bruitées ou comportant des points aberrants dues aux limitations des capteurs ;

Mélange En dépit de la résolution des images fournies, il arrive qu'un même pixel contienne plusieurs réponses spectrales. Il est alors difficile d'affecter ces pixels à une seule classe ;

Superposition de classes De même, il arrive souvent que les classes de pixels se chevauchent dans l'espace des variables. Bien que deux pixels appartiennent à des classes différentes, ils peuvent néanmoins avoir des attributs très similaires ;

Nombre de classes Le nombre de classes peut rarement être connu préalablement. Il

n'y a en effet aucune raison de privilégier a priori un nombre de classes plutôt qu'un autre ;

Densités de classe inégales Les classes ont de fortes chances d'être de densités très différentes (on appelle densité, le nombre de données par unité de volume dans l'espace des données) ;

Tailles de classe inégales Les effectifs des classes n'ont par ailleurs aucune raison d'être similaires ;

C'est donc dans ce cadre particulier que se situe la problématique de notre contribution. Notre travail s'inscrit dans la continuité de celui effectué en classification dans notre laboratoire ces dernières années. Une méthode de classification automatique non supervisée, basée sur l'estimation de la fonction de densité de probabilité a été développée [94][95]. Cette méthode présente l'avantage de ne pas faire d'a priori sur la forme des classes. Autrement dit, et contrairement, par exemple, aux méthodes de type k-means, elle permet de détecter des classes de forme quelconque, non nécessairement convexes. Elle repose sur une estimation de la fonction de densité de probabilité en tout point de l'espace des données, puis sur la détection des frontières entre les classes à l'aide d'algorithmes de type "watersheds". Elle s'avère particulièrement efficace pour la segmentation d'images multicomposantes. Ses principaux inconvénients sont : la nécessité de réduire la dimensionnalité à 3, et le temps de calcul engendré par l'estimation de la densité en chaque point de l'espace.

G. Cutrona et N. Bonnet ont développé une extension floue [51] de la méthode Herbin-Bonnet-Vautrot décrite avant. Les extensions floues apportent une finesse, une flexibilité et une efficacité non-négligeables aux méthodes de classification "classiques". Le plus connu des algorithmes de classification floue est vraisemblablement l'algorithme des *Fuzzy C-Means* (ou FCM) [10]. Cet algorithme est notamment très utilisé pour la segmentation d'images. Il est bien adapté pour représenter le caractère ambigu de certains pixels.
Le principe des méthodes de classification floue est de représenter les classes comme des sous-ensembles flous. Ainsi, à la fin du processus de classification, chaque donnée de l'échantillon initial appartient à toutes les classes avec des degrés d'appartenance différents. Le degré d'appartenance floue d'un point à une classe est une valeur comprise entre 0 et 1. En utilisant un abus de langage on pourrait dire autrement que plus une donnée appartient à une classe, et plus son degré d'appartenance à cette classe est proche de 1. Inversement moins une donnée appartient à une classe et plus son degré d'appartenance à cette classe est proche de 0. Dans les approches non floues, une donnée appartient ou n'appartient pas à une classe.
L'adjonction de flou dans les méthodes de classification permet d'ajouter de la finesse au processus et facilite l'interprétation des résultats. Outre l'imprécision et l'incertitude inhérentes à l'affectation des données qu'il permet de modéliser, le flou permet aussi de

représenter le chevauchement (ou le mélange) de classes.
L'utilisation du flou pour la définition des classes présente de nombreux avantages et utilise cependant des données qui ne sont pas floues.

Notre idée est donc d'apporter ce flou dès la représentation des données, avant même de chercher les classes.
Nous avons pour cela développé une nouvelle façon de représenter les données et les liens entre les données. Notre contribution a consisté à élaborer une nouvelle manière de représenter les données et l'ensemble des données en utilisant le flou. Une étape préalable de transformation par rang, nous permet d'obtenir par la suite une méthode robuste et permet de résoudre le problème des classes de densités inégales. Nous définissons ensuite chaque donnée comme un sous-ensemble flou (ce qui peut entre autre permettre de prendre compte l'imprécision liée au système d'acquisition), puis l'ensemble complet lui-même comme un ensemble flou. Le degré d'appartenance à l'échantillon est le résultat d'un processus d'agrégation et constitue en quelque sorte une fonction de score sur les données, dont l'interprétation est proche de la notion de fonction de densité de probabilité.

Après cette phase de représentation des données, nous proposons une nouvelle notion baptisée "connectivité", qui pourra être utilisée dans un processus de classification. Cette approche nous place donc très en amont des méthodes de classification proprement dites. La notion de connectivité exprime la manière avec laquelle un point en connecte un autre à l'échantillon. Cette notion de liaison entre les données complète les précédentes consacrées à la représentation de ces données.

Cette dernière étape place donc les algorithmes issus de ce système de représentation, dans la catégorie des méthodes dites "density-based connectivity" [8] (voir section 4.2).
Notre système de représentation a donné lieu à plusieurs communications : un première sur une version simplifiée [20] de notre méthode puis une seconde sur le système global [13]. D'autres travaux sont engagés sur ce thème.

Un exemple didactique nous permettra de présenter les idées sous-jacentes de manière plus intuitive. Il faut par ailleurs noter que ces nouvelles notions, plongées dans le cadre de la théorie des ensembles flous, peuvent également être exprimées sous d'autres formes, comme par exemple en termes empruntés à la théorie de la décision et du choix [138][119]. Ces façons différentes de voir permettent également de mieux appréhender l'intérêt et les nombreuses possibilités de ce travail.

Dans la section 4.2, nous décrirons les grandes familles de méthodes de classification et en présenterons les plus importantes. Dans la partie suivante (4.3), après avoir présenté un exemple didactique introductif, nous décrirons les trois éléments de la nouvelle représentation des données pour la classification que nous proposons, ainsi que des exemples et des applications. Enfin des éléments de discussion et de conclusions seront exposés dans la section 4.5.

4.2 Survol des méthodes de classification

Il existe de nombreux ouvrages consacrés (où comportant de larges chapitres) à la classification. Les ouvrages et articles d'Hartigan [87], Jain et Dubes [111], ou Benzecri [7] sont des références historiques. On pourra se référer aux livres -en langue française- de Lebart et al [135], Saporta [167], ou Celeux et al [39] qui présentent la classification, parmi d'autres outils, dans le cadre plus général de l'analyse de données. Des articles plus récents présentent et comparent les algorithmes de classification les plus classiques et les plus efficaces [8][112].

Nous adopterons dans cet exposé, un système de classement proche de celui retenu par Berkhin [8]. La suite de cette section va donc présenter les différentes catégories de méthodes de classification :

- méthodes hiérarchiques
  - algorithmes ascendants (ou agglomératifs)
  - algorithmes descendants (ou divisifs)
- méthodes de partitionnement
  - méthodes des nuées dynamiques
  - algorithmes basés sur la densité

4.2.1 Méthodes hiérarchiques

Les méthodes hiérarchiques [148] construisent des hiérarchies de partitions. Autrement dit, le résultat d'une méthode hiérarchique peut être vu comme un arbre de classes. On appelle ce type de représentation un dendrogramme. La figure 4.1 représente, sur le graphique du bas, le dendrogramme obtenu par classification hiérarchique (en utilisant la distance du saut minimum dont nous donnerons la définition après) des données représentées dans le plan, sur le graphique du haut. Toute coupe de l'arbre fournit une partition, et donc une classification. La structure d'arbre induit par ailleurs une relation d'inclusion entre des classes déterminées à différentes profondeurs d'une même branche. Ce type de résultat permet d'effectuer des classifications avec différents degrés de granularité.
On distingue les méthodes ascendantes des méthodes descendantes [111][118][30]. Les

méthodes ascendantes débutent avec des classes d'un seul élément (singletons) puis, récursivement, agglomèrent les classes. Les méthodes descendantes commencent avec une classes correspondant tout l'ensemble des données et séparent récursivement les classes appropriées. Le processus se termine en fonction d'un critère d'arrêt donné (ce critère peut être par exemple un nombre donné de classes à trouver) ou bien détermine toute la hiérarchie. Dans la suite de cet exposé, pour ne pas en alourdir la lecture, nous ne traiterons que le cas des méthodes ascendantes, le cas divisif se déduisant aisément du cas agglomératif.

Le principe des méthodes ascendantes hiérarchiques consiste à créer à chaque étape une partition de l'ensemble des données en agrégeant deux à deux les éléments les plus proches. Lors de la première itération les éléments sont les données initiales (singletons) puis dans les étapes suivantes, ils peuvent être soit un regroupement de données, soit les données elles-même. Comme nous l'avons dit avant, les résultats sont fournis sous forme de hiérarchie de partitions et représentés sous forme d'arbre dont chaque "élagage" fournit une partition de l'ensemble des données.
Dans ce type de méthodes, les éléments à agrégés sont choisis de sorte qu'ils soient les plus proches. Il est donc nécessaire de disposer d'une distance pour quantifier cette proximité. Lors de la première étape, cette distance est une distance sur l'espace contenant les données. Il suffit donc que l'espace soit muni d'une métrique. Lors des étapes suivantes, les proximités sont calculées entre groupements d'éléments. On utilise donc des généralisations de la notion de distance à des sous-ensembles de points. La règle de calcul des distances entre des groupes disjoints de données est appelée *critère d'agrégation*. On calcule généralement cette distance entre groupements de données à partir des distances entre les éléments des groupements impliqués. On peut écrire cette distance de façon générale de la manière suivante :

$$d(C_i, C_j) = operation\{d(x, y) | x \in C_i, y \in C_j\}$$

où C_i et C_j sont deux groupements d'individus et où *operation* désigne l'opérateur utilisé sur l'ensemble des distances entre les paires d'éléments des groupements.
Les opérations les plus courantes sont [39] :

- le minimum : $d(C_i, C_j) = \min\{d(x, y) | x \in C_i, y \in C_j\}$ (on appelle cette distance le saut minimal *single linkage*)
- la moyenne : $d(C_i, C_j) = moyenne\{d(x, y) | x \in C_i, y \in C_j\}$ (on appelle cette distance la distance moyenne *average linkage*)
- le maximum : $d(C_i, C_j) = \max\{d(x, y) | x \in C_i, y \in C_j\}$ (on appelle cette distance le saut maximal *complete linkage*)

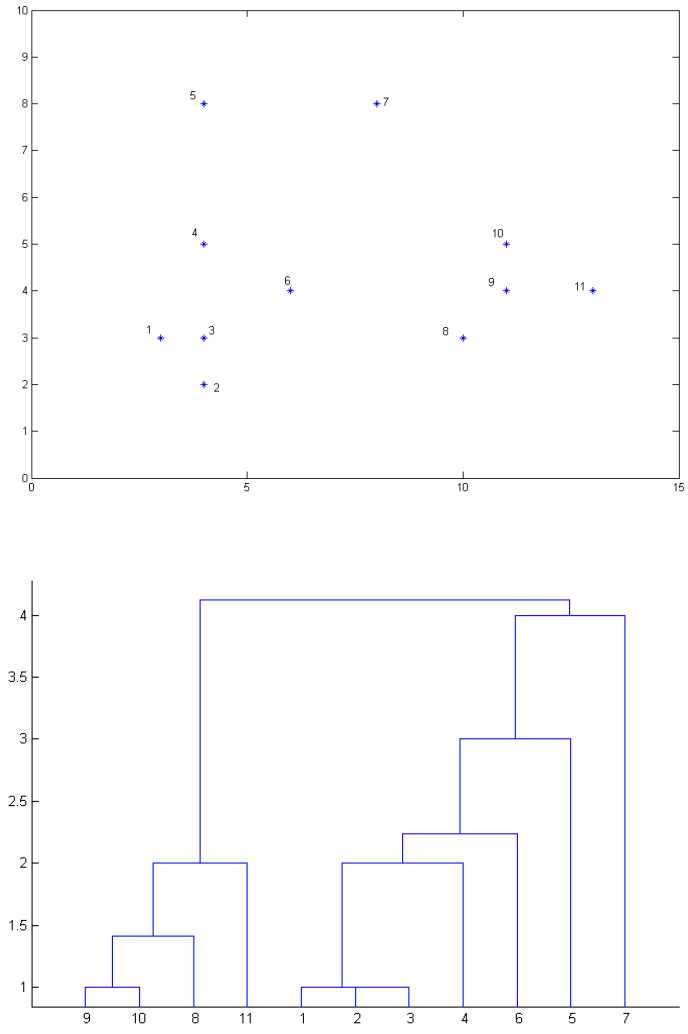


FIG. 4.1 – Exemple de dendrogramme : Echantillon (haut) et dendrogramme (bas) associé à la classification hiérarchique des données correspondantes. Les nombres indiqués en abscisse du dendrogramme correspondent à la numérotation des points de l'échantillon.

L'algorithme fondamental utilisé pour la classification ascendante hiérarchique est le suivant :

```
0 : Ensemble de  n éléments à classer
1 : Construction de la matrice des distances entre les éléments
2 : Recherche des deux éléments les plus proches
3 : Agrégation de ces deux éléments en un nouvel élément
4 : Construction de la nouvelle matrice des distances
        - suppression des éléments ayant été agrégés
        - mise à jour de la matrice en calculant les distances du nouvel
          élément résultant de l'agrégation avec tous les autres
3 : Retour en 2 jusqu'à ce qu'il n'y ait plus qu'un élément
    constitué de toutes les données
```

Remarque : On constate que l'algorithme utilise le classement des couples d'éléments, induit par les distances. Ce classement est appelé *ordonnance* (ou *préordonnance* s'il y a des égalités).

Le premier avantage des méthodes hiérarchiques est la flexibilité induite par le système de granularité des classes sus-cité. Ensuite il est possible de choisir parmi un grand nombre de dissimilarités ou de distances. Par conséquent, ce type de méthodes est utilisable avec n'importe quel type de données. Les inconvénients majeurs de cette famille d'algorithmes sont :

- le choix du critère d'arrêt n'est pas simple
- la plupart des algorithmes ne revisitent pas les classes une fois qu'ils ont été construits et ne peuvent donc pas être améliorés
- si le choix de la métrique est vaste il n'est en revanche pas simple

Remarque : Il est équivalent de munir un ensemble de données d'une ultramétrique (une distance d est dite *ultramétrique* si $d(x,y) \leq \{d(x,z), d(z,y)\}$) ou de chercher une hiérarchie indicée de parties de cet ensemble. Une hiérarchie indicée H est une hiérarchie (i.e. une famille de partie de l'ensemble contenant l'ensemble entier ainsi que les singletons et telle que les autres parties de H sont soit distinctes soit incluses l'une dans l'autre) telle que, à toute partie h de H, on peut associer une valeur numérique $v(h) \geq 0$ qui vérifie : $h \subset h' \Rightarrow v(h) < v(h')$.

Remarque : A partir de la matrice de distance entre les données, on peut créer un graphe dont les sommets sont les données et les poids des arrêtes sont les éléments de la

matrice correspondants. Cette association fait ainsi le lien entre classification hiérarchique et partitionnement de graphes. La recherche d'un arbre couvrant minimal (*minimal spanning tree*) de ce graphe donne un résultat équivalent à celui obtenu par une classification ascendante utilisant la distance du saut minimal [76].

SLINK [173], la méthode de Voorhees [193] et CLINK [52] sont des exemples d'algorithmes utilisant respectivement les métriques du minimum, de la moyenne et du maximum.
En dépit du coût en temps de calcul élevé, les méthodes hiérarchiques sont très utilisées. L'algorithme AGNES (AGlomerative NESting) [118] est notamment très populaire. Il existe une implémentation de cet algorithme dans un package destiné au logiciel `R` (et initialement conçu pour `Splus`) à l'adresse `http://cran.cict.fr/src/contrib/Descriptions/cluster.html`.

Les métriques de liens basées sur la distance euclidienne conduisent naturellement à construire des classes de forme convexe. Comme nous l'avons vu, dans le cas d'images multicomposantes (et plus généralement dans le cas de données spatiales), les classes ont des formes quelconques et donc non nécessairement convexes. L'algorithme CURE (Clustering Using REesentatives) [84] apporte des solutions pour les problèmes de forme des classes et de sensibilité aux données aberrantes. Le principe consiste à utiliser, pour chaque classe un nombre c (donné) de points appelés représentants. Les distances sont alors calculées en utilisant ces représentants (avec le lien minimum). Cette approche est un compromis entre les méthodes utilisant tous les points (méthodes de graphes) et les méthodes utilisant les centres des classes. Les représentants initiaux sont choisis proches des centres "géométriques" des classes (en utilisant un paramètre défini par l'utilisateur). Les points aberrants n'influencent ainsi pas le choix des représentants. Il faut noter par ailleurs que cet algorithme utilise également un partitionnement initial en p partitions, donnant ainsi une première granularité à la partition.

4.2.2 Méthodes de partitionnement

Dans cette partie nous décrivons les méthodes non-hiérarchiques consistant à partitionner l'ensemble des données, c'est à dire qui divisent l'ensemble des individus en sous-ensembles. Puisqu'il est impossible de tester toutes les combinaisons possibles, différentes approches sont disponibles pour résoudre ce problème. Les plus classiques méthodes sont des méthodes dites de réallocation. Les *k-means* sont sans doute la technique la plus connue. Ces méthodes consistent à déterminer les classes en réaffectant itérativement les points aux classes. L'autre approche que nous avons choisie d'évoquer ici consiste à estimer la fonction de distribution des données. Autrement dit, il s'agit d'estimer la fonction de densité de probabilité sous-jacente. Ce type de techniques permet de ne pas

faire d'hypothèse sur la forme des classes et donc de pouvoir trouver des classes de forme quelconque (i.e. non convexe).
Nous allons donc exposer les méthodes de nuées dynamiques puis, dans une seconde sous-partie, les méthodes basées sur l'estimation de la densité de probabilité.

4.2.2.1 Méthodes de réallocation

Les méthodes de réallocation contiennent essentiellement deux algorithmes (et leurs variantes) classiques : la technique d'agrégation autour des centres mobiles (ou méthode de Forgy) et les nuées dynamiques [57][58]. Dans le premier algorithme, les classes sont représentées par leurs centres, tandis que dans le second, elles sont représentées par un ensemble de points choisis et appelés "étalons".

L'algorithme d'agrégation autour des centres mobiles nécessite tout d'abord que l'espace des données soit muni d'une distance. On considère un ensemble de n données décrites par p variables. On suppose également qu'il n'existe pas plus de q classes dans cet échantillon. L'algorithme s'écrira alors :

```
0 : on détermine q centres temporaires
    ces centres induisent une première partition de
    l'ensemble des données en q classes, chaque
    donnée étant affectée à la classe du centre
    qui lui est le plus proche

1 : on détermine q nouveaux centres en prenant les
    centres de gravité des classes obtenues à
    l'étape 0
    ces centres induisent une nouvelle partition
    obtenue comme en 1

on réitère jusqu'à stabilisation du processus
```

Le processus se stabilise nécessairement (voir par exemple [135]) et l'algorithme s'arrête lorsque la même partition est obtenue deux fois successivement (ou lorsqu'un critère d'arrêt acceptable est atteint). Le résultat obtenu dépend du choix initial fait à l'étape 0. On se reportera à [140] pour des recommandations quant à ce choix.

Dans la technique des nuées dynamiques, considérée comme une généralisation de l'algorithme précédent, les centres sont remplacés par plusieurs individus de la classe

appelés “étalons” et permettant de mieux décrire les classes.

La technique des $k-means$ est légèrement différente de l'agrégation autour des centres mobiles [140]. Ele débute par un tirage aléatoire des centres mais la détermination des nouveaux centres se fait différemment. En effet, dans les $k-means$, chaque réaffectation d'un individu conduit à une modification de la position du centre correspondant. La convergence peut être plus rapide mais dépend de l'ordre dans lequel sont réaffectées les données.

Même si des choix judicieux peuvent être faits pour les centres initiaux, généralement, des initialisations différentes donneront des résultats différents. Il existe donc une méthodologie consistant à effectuer plusieurs choix et de tester la stabilité des résultats. Si les partitions changent complètement à chaque fois, il est difficile d'en déduire une partition pertinente. Dans le cas contraire, on peut supposer que les classes sont réalistes. On appelle forme forte, un ensemble d'éléments qui n'a pas été séparé par les diverse partitions générées. Autrement dit, la recherche des formes fortes consiste à chercher les sous ensembles de données qui ont été affectés aux mêmes classes lors des partitions successives obtenues avec les tirages de centres différents.
En pratique, la recherche des formes fortes ne fournit pas une classification exploitable. Le nombre de classes est en effet généralement beaucoup trop élevé. De plus une grande partie des classes est d'effectif trop faible. Cependant, cette recherche peut permettre de suggérer un nombre de classes à chercher en éliminant les classes d'effectifs trop faibles et en réaffectant leurs éléments aux classes conservées.

Coupler les méthodes hiérarchiques aux méthodes de réallocations permet d'obtenir des techniques relativement efficaces appelées méthodes mixtes [198]. Le principe consiste à partitionner une première fois l'ensemble des données à l'aide d'un algorithme comme l'agrégation autour des centres mobiles, puis d'appliquer une méthode hiérarchique sur les groupes obtenus. Une nouvelle utilisation des centres mobiles peut enfin permettre d'optimiser les partitions suggérées par la méthode hiérarchique et l'analyse de son dendrogramme.

Le grand inconvénient des méthodes de réallocation est qu'elles privilégient les classes de forme sphérique. L'utilisation de distances adaptatives permet d'améliorer un peu les méthodes mais les classes trouvées restent généralement de forme ellipsoïdale et ne parviennent pas à détecter les classes non convexes.

Les méthodes basées sur la densité de probabilité que nous allons présenter dans la section suivante ne présentent pas cet inconvénient.

4.2.2.2 Méthodes basées sur la densité de probabilité

L'idée de base sur laquelle repose ce type de méthodes est de définir les classes comme des zones denses connexes. La forme des classes est ainsi liée à la fonction de densité sous-jacente. Il n'y a donc pas d'hypothèse sur la forme des classes comme dans les méthodes de réallocations. L'estimation de la fonction de densité de probabilité est un problème bien connu en statistiques et en analyse de données, et peut se faire selon deux approches différentes [174][170].

Estimation de la fonction de densité de probabilité. On considère un ensemble de n points $x_1, x_2, \ldots, x_n$ de $\mathbb{R}^p$ dont on souhaite estimer la densité de probabilité sous-jacente. On appelle $\hat{f}$ cette estimation. Il existe principalement deux façons (non paramétriques[2]) de calculer $\hat{f}$. La première est une technique de k plus proches voisins et la seconde est basée sur les noyaux [72][64][56].

La méthode à noyaux de Parzen [155] estime la densité en un point en utilisant les données contenues dans un volume (la fenêtr de Parzen) centré sur ce point. Elle nécessite aussi la définition d'une fonction noyau qui permet en quelque sorte de de traduire la contribution d'un point à la densité estimée, en fonction de sa distance au point où elle est calculée. L'estimation de la densité en un point x est donnée par :

$$\hat{f}_h(x) = \frac{1}{n.h^p}.\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{4.1}$$

où h est la taille de la fenêtre de Parzen, $||.||$ la norme euclidienne sur $\mathbb{R}^p$ et K la fonction noyau.
Une fonction K de $\mathbb{R}^p$ dans $\mathbb{R}$ est une fonction noyau s'il existe une fonction $g : [0, \infty[\rightarrow \mathbb{R}$

- positive
- non croissante
- continue par morceaux
- et telle que $\int_0^\infty g(t)dt < \infty$

de sorte que :

$$K(x) = g(||x||^2)$$

Le choix de la fonction noyau est donc large et interfère peu sur les résultats de classification effectuée ensuite. On peut néanmoins citer [42] deux noyaux très fréquemment utilisés :

[2] il existe également des méthodes paramétriques et semi-paramétriques d'estimation mais ce n'est pas notre propos ici

– le noyau uniforme

$$K(x) = \begin{cases} 1 \text{ si } ||x|| \leq 1 \\ 0 \text{ si } ||x|| \geq 1 \end{cases}$$

– le noyau Gaussien

$$K(x) = e^{-||x||^2}$$

La fonction obtenue est elle-même une densité. Elle hérite des propriétés de continuité et de dérivabilité de la fonction noyau.
Le choix du paramètre h (la taille de la fenêtre de Parzen) est important et même crucial. Ce paramètre de lissage [194] doit être choisi avec soin : trop grand, et l'estimation risque de masquer certaines variations de la "vraie" fonction de densité sous-jacente, trop petit et l'estimation risque d'être parasitée et laisser apparaître des variations n'existant pas. Le défaut majeur de cette estimation en est une conséquence : les queues de distribution apparaissent très bruitées dans le résultat de l'estimation. Si on lisse suffisamment pour que cela n'arrive pas, des détails essentiels de la partie principale de la distribution disparaissent.

La méthode basée sur les k plus proches voisins est sensiblement différente. Cette fois, le nombre de données de l'échantillon utilisées pour estimer la densité en un point est fixé (et le volume les contenant est donc variable, ce qui était le contraire dans la technique de Parzen). L'estimation est donnée par l'expression :

$$\hat{f}_k(x) = \frac{k}{2.n.d_k(x)} \tag{4.2}$$

où k est le paramètre, et $d_k(x)$ la distance de x à son $k$ième plus proche voisin.
Le lissage est ici en quelque sorte "adaptatif". L'inconvénient de cette estimation est que sa dérivée n'est pas continue. Contrairement à la fonction obtenue avec la méthode de Parzen, l'estimation fournie ici n'est pas une densité (son intégrale sur l'espace entier ne vaut pas 1). Par ailleurs, les queues de distribution ont tendance à diminuer très lentement. Cette estimation n'est donc pas adaptée lorsqu'une estimation "complète" est requise.

Examinons maintenant quelques uns des plus efficaces algorithmes basés sur la densité. Certains utilisent directement les estimations que nous venons de présenter, d'autres introduisent des notions très proches.

DBSCAN. L'algorithme le plus utilisé est sans doute DBSCAN (ou son extension GDBSCAN) [67][166]. Cette méthode introduit deux notions : l'accessibilité et la connexité, au sens de la densité (*density-reachable* et *density-connectivity*). DBSCAN utilise deux paramètres d'entrée, $MinPts$ et Eps, et définit :

1. un voisinage de taille Eps d'un point x de l'ensemble X des données :

$$N_{Eps}(x) = \{y \in X | d(x,y) \leq Eps\}$$

2. un noyau est un point ayant plus de $MinPts$ points dans son voisinage
3. un point y est accessible, au sens de la densité, d'un point noyau x s'il existe une séquence finie de points noyaux entre x et y tels que chaque point est dans le voisinage de son prédecesseur
4. deux points x et y sont connexes, au sens de la densité, s'ils sont accessibles, au sens de la densité, d'un point noyau commun

L'algorithme consiste alors à déterminer les ensembles de données connexes qui forment les classes. Les points qui ne sont connectés à aucun point noyau sont considérés comme des points aberrants ou du bruit. L'estimation des paramètres, comme dans pratiquement toutes les méthodes de classification, n'est pas simple. Les auteurs de DBSCAN suggèrent de fixer le paramètre $Minpts$ à 4. Cette valeur semble être, d'après l'expérience des auteurs, la valeur la plus adaptée aux bases de données de dimension 2. Pour l'estimation de Eps, deux solutions sont proposées : l'une interactive et l'autre basée sur l'étude d'une courbe particulière. Cette courbe consiste à ordonner les points de la base de données sur l'axe des abscisses en fonction de leur k-distance (i.e. la distance qui les séparent de leurs $k$eme plus proches voisins respectifs) et, en ordonnée, les valeurs des k-distances correspondantes. Le point correspondant à la première "vallée" de cette courbe définit un seuil qui détermine Eps. Ces choix acrobatiques illustrent la difficulté systématiquement rencontrée dans les méthodes de classification automatique : le choix des paramètres.

Remarque : Il existe une version améliorée de DBSCAN, et baptisée OPTICS (Ordering Points To Identify the Clustering Structure) [2], qui permet de déterminer ces paramètres de manière plus efficace (au prix de l'introduction de nouvelles notions).

Cette méthode utilise, d'une manière un peu "détournée" une estimation par k-plus proches voisins.

DENCLUE L'approche de Hinneburg et Keim est assez différente. Leur méthode de classification appelée DENCLUE (DENsity based CLUstEring) [99] repose sur une estimation de la densité sur tout l'échantillon définie comme la somme de fonctions d'influences (proche de la notion de fonction noyau) calculée sur chacun des individus de l'échantillon. La classification est ensuite effectuée en introduisant une notion "d'attracteur de densité" (correspondant à des maxima locaux de la densité). Ces attracteurs sont calculés avec une méthode de montée du gradient. Chaque attracteur et les points qui lui

sont rattachés définissent les classes. Ce processus utilise deux paramètres caractérisant respectivement l'influence d'un point dans l'espace des données, et le caractère significatif ou non d'un attracteur.
De nombreux algorithmes peuvent être vus comme des cas particuliers de DENCLUE. En prenant une fonction d'influence uniforme et des paramètres égaux à Eps et $MinPts$, les classes correspondent à celles définies dans l'algorithme DBSCAN. Par ailleurs on peut constater une grande similarité dans l'approche avec la classification par *Mean Shift* que nous allons décrire.

Méthode par Mean-Shift Le *Mean Shift* est une procédure itérative qui associe à chaque point de l'échantillon la moyenne des points de son voisinage. Cette procédure a été utilisée pour l'analyse de données [174][73]. Dans [42] Cheng généralise le *Mean Shift* de différentes manières : il généralise la procédure à des noyaux non plats, introduit la pondération des données et propose d'utiliser la procédure dans des sous-ensembles de l'échantillon initial. Comaniciu et col. [45] ont réutilisé le mean shift comme pour procédure de classification, pour la segmentation d'images.
Si on considère un échantillon $X = \{x_1, x_2, \ldots, x_n\}$ de n points de $\mathbb{R}^p$, et une fonction noyau K (voir avant), alors le mean shift en un point x de $\mathbb{R}^p$ est défini par :

$$M_{h,K}(x) = m(x) - x \tag{4.3}$$

où :

$$m(x) = \frac{\sum_{i=1}^{n} K(x_i - x)x_i}{\sum_{i=1}^{n} K(x_i - x)} \tag{4.4}$$

La procédure du mean shift correspond à un méthode de montée du gradient (on peut trouver une méthode utilisant le gradient et la densité dans [129]) de la fonction de densité de probabilité. Cette correspondance est démontrée dans [42] ou [45] en établissant que l'amplitude du vecteur du mean shift est proportionnelle au ratio du gradient de la densité sur la densité estimée :

$$M_{h,K} = \frac{1}{2}h^2 c. \frac{\hat{\nabla} f_{h,K}(x)}{\hat{f}_{h,K}(x)} \tag{4.5}$$

(où c est une constante, $\hat{\nabla} f_{h,K}$ est l'estimation du gradient (calculé comme le gradient de l'estimation de a densité)) Le vecteur du mean shift suit donc l'accroissement maximum de la densité estimée. Ce constat est finalement assez intuitif. En effet, la moyenne se "dirige" vers les régions contenant le plus de points. On retrouve d'ailleurs cette idée dans les algorithmes de type *k-means*.
La procédure du Mean Shift, qui consiste à :

1. calculer le vecteur du mean shift au point x
2. déplacement (translation) du noyau suivant le vecteur calculé

converge alors [42] vers un point annulant le gradient et donc à un maximum local de la densité estimée. Par ailleurs dans les régions de faible densité, les déplacements sont beaucoup plus larges. De même les translations sont plus petites aux environs des maxima locaux, où l'analyse est donc plus fine. La procédure du mean shift peut être donc considérée comme méthode de montée de gradient adaptative [45].
En lançant une procédure de mean shift sur chaque point de l'échantillon intial, on associe à chacune de ces données, un mode (i.e. un maximum local), résultat de la convergence de la procédure. Chaque mode induit donc une classe de données (un regroupement des modes très proches -via un paramètre de tolérance- est en pratique nécessaire). Comaniciu propose une application de cette méthode pour la segmentation d'images couleur avec de très bons résultats [46]. La sélection du paramètre de lissage h est toujours la difficulté de ce type de méthode. Une solution est proposée dans [44] pour l'estimation de ce paramètre, étudiant pour chaque point la matrice de covariance la plus stable à travers une échelle de valeurs du paramètre. Cette technique de classification est plutôt efficace mais n'échappe pas à la malédiction de la dimensionnalité.

Nous avons implémenté la technique du mean shift pour la segmentation d'image couleur (voir [49] pour plus d'informations sur la segmentation). L'algorithme a été testé sur une image couleur (figure 4.2) classique et les résultats, pour différents paramètres de lissage, sont présentés sur la figure 4.3.

FIG. 4.2 – Image couleur "Lena"

Méthode de M. Herbin et al. La méthode classification développée par M. Herbin, N. Bonnet, P. Vautrot et J. Cutrona [94][95][51] utilise différemment la densité de probabilité estimée. Le principe de cette méthode développée depuis 1996 au laboratoire consiste tout d'abord à discrétiser l'espace dans lequel sont les données. La fonction de densité de probabilité est ensuite estimée *en chaque point* de cet espace discrétisé. La

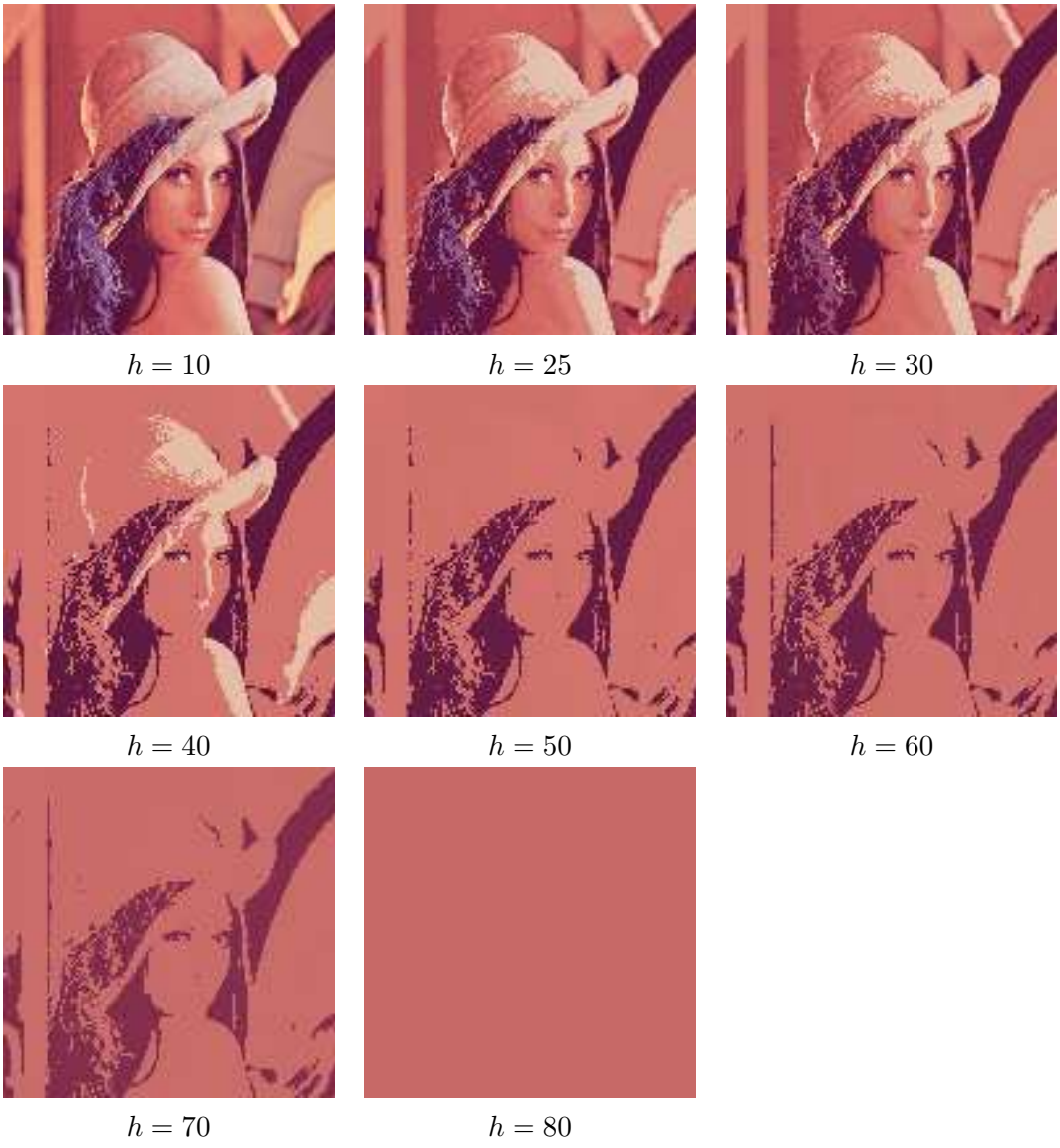


FIG. 4.3 – Segmentation de l'image "Lena" à l'aide de l'algorithme du Mean-Shift avec différents paramètres de lissage. Les couleurs des classes (régions) sont les couleurs moyennes des pixels composant ces classes.

densité est donc complètement déterminée (relativement aux paramètres de discrétisation). Elle est estimée en utilisant la méthode de Parzen.
Chaque mode de la densité estimée induit une classe de données. Ces classes sont établies en utilisant une méthode de lignes de partage des eaux (ou *watersheds* [162]). L'utilisation de cette technique présente l'avantage de permettre de déterminer avec finesse les frontières entre les classes. Ce point est en général assez délicat pour les autres méthodes de classification.
Le paramètre de lissage -problème récurrent- est estimé à l'aide de techniques de bootstrap [96].
Lorsque la dimensionnalité est trop élevée, une réduction est effectuée à l'aide, par exemple d'une ACP. L'inconvénient de cette technique est la lourdeur engendrée par le calcul "exhaustif" (i.e. sur tout l'espace) de l'estimation de la densité.
Une extension floue efficace de la méthode a été développée [51] et de nombreuses applications ont été présentées.

4.2.3 Méthodes de classification floue

Le principe des méthodes de classification floue consiste à définir les classes comme des sous-ensembles flous [10][100]. Chaque donnée appartient alors à toutes les classes avec différents degrés. Ce principe est donc, par définition, beaucoup moins rigide que l'approche classique (*crisp*).
De nombreuses extensions floues basées sur les méthodes classiques ont été développées mais la plus connue et la plus utilisée reste l'algorithme dit des *Fuzzy C-means* [10], utilisé notamment en segmentation d'images [25][164].
Cette classe de méthodes fait l'objet de recherches nombreuses et nous semble être une approche riche et la plus adaptée aux types de données que nous manipulons.

4.2.4 Conclusion

Bien que la section précédente n'ait présenté que quelques techniques et approches de classification[3], on peut constater que le choix est large. Comme nous l'avons expliqué au début de ce chapitre, notre but principal est d'effectuer de la classification de pixels d'images multicomposantes. Toutes les méthodes ne sont pas adaptées aux contraintes spécifiques de ce type de données (voir l'introduction). Les méthodes hiérarchiques ne sont pas forcément adéquates compte tenu de la quantité généralement énorme d'individus à classer. Par ailleurs la forme arbitraire justifie l'emploi de méthodes basées sur la densité.

[3] Le but n'était pas ici de faire une liste descriptive exhaustive des méthodes de classification mais d'exposer différents types d'approches. Pour des listes plus complètes et plutôt récentes sur le sujet on se reportera à [112] ou [8]

Nous nous sommes attachés, dans le travail présenté à la partie suivante, à développer un nouveau mode de représentation des données, en amont du processus de classification intégrant les avantages suivants :

- robustesse vis-à-vis des points aberrants
- détection de classes de densités très différentes
- l'utilisation du flou pour représenter l'imprécision des valeurs liées aux données (pixels)
- d'une notion proche de celle de densité pour la détection de classes de forme quelconque

Ce nouveau type de représentation peut s'intégrer dans des processus de classification classique. On peut noter des points avec les approches des algorithmes Herbin et al., DBSCAN ou DENCLUE.

Nous venons donc de voir qu'il existait différentes approches pour la classification. Comme nous l'avons déjà dit dans l'introduction, plusieurs de ces méthodes possèdent des extensions effectuant une classification floue. Le flou, dans ce cas apparaît à la fin du processus, sur les classes. Nous expliquer comment mettre en place un système de représentation (en quelque sorte complémentaire) qui introduit du flou sur les données à classer. Nous allons tout d'abord détailler la méthode puis une utilisation pour la classification sera présentée. Nous exposerons également des exemples d'application puis une discussion et une conclusion seront proposées.

4.3 Une nouvelle représentation floue des données d'un échantillon

Nous allons maintenant présenter notre technique de représentation. Comme nous l'avons déjà dit, nous utilisons la théorie des sous-ensembles flous [205][23][24] avant le processus de classification, pour représenter les données et les liens entre ces données. Après une transformation du tableau de dissimilarité entre les données, nous définissons les données comme des sous ensembles flous. Ces ensembles flous sont un peu particuliers dans la mesure où les fonctions d'appartenance sont des fonctions des rangs des données. Ces sous-ensembles flous sont ensuite agrégés et nous permettent de représenter l'échantillon en entier comme un sous-ensemble flou. Enfin on définit à partir de ces informations une notion de connectivité. Cette notion se traduit pour chaque point de l'échantillon par un ensemble de connexion. L'ensemble de connexion d'un individu est un sous-ensemble flou de l'échantillon. Cet ensemble d'outils de représentation constitue un système robuste et bien adapté aux contraintes de nos données.

Nous allons tout d'abord introduire un exemple didactique qui permet de mieux appréhender les idées sous-jacentes. Nous exposerons ensuite en détails les notions introduites avant de proposer une application de classification.

4.3.1 Exemple introductif

Supposons que nous ayons un ensemble d'individus caractérisés par leurs opinions sur un thème donné (politique, économique, artistique...) (voir figure 4.4) et que nous devions classer ces individus en groupes pertinents. On suppose par ailleurs que chaque individu peut comparer son opinion avec celle des autres.

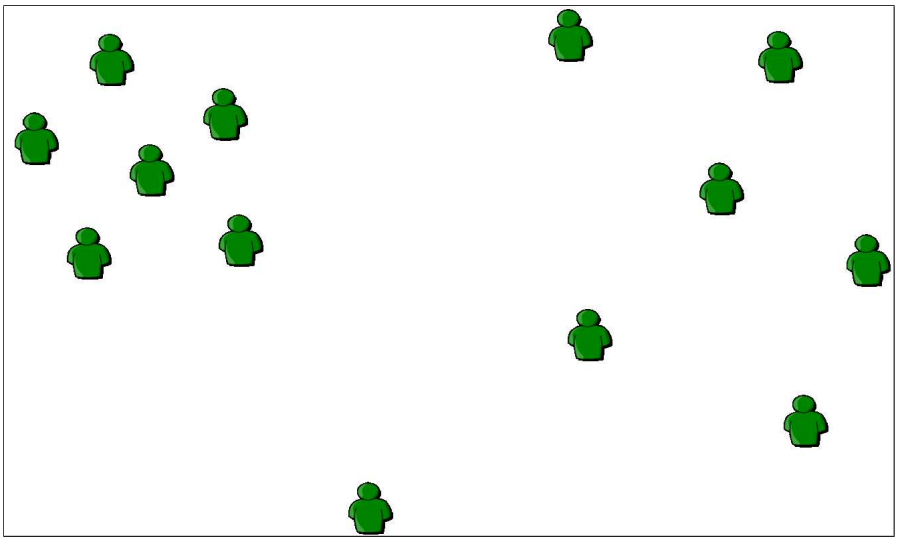


FIG. 4.4 – Exemple introductif : Echantillon de personnes étudiées ; les individus sont caractérisés par leur opinion sur un sujet donné.

Notre première idée consiste à déterminer, pour chaque individu, un classement de tous les autres, en fonction de ses affinités relatives à ses opinions. Autrement dit on dispose pour chaque individu de ses préférences sur l'ensemble de l'échantillon. On décide alors pour chaque individu d'attribuer à tous les autres des degrés dépendant du classement défini par cet individu. Ce degré quantifie donc en quelque sorte la notion de "plus préféré". Prenons par exemple un individu au hasard dans cet échantillon. On dispose pour cet individu d'un classement de tous les autres au regard de leurs opinions. On attribue à l'individu classé premier (i.e. son préféré) un degré par exemple égal à 0.9

puis au second préféré, un degré valant 0.8, 0.7 au troisième etc... et 0 à celui qui préfère le moins.
L'élément important est que ce degré dépend de la position dans le classement et non de la proximité entre les opinions. Considérons par exemple un individu marginal dont les opinions sont très différentes de celles de tous les autres. Cet individu induit tout de même un classement de tous les autres par ordre d'affinité d'opinion et on attribue un degré de 0.9 à l'individu classé premier. Les degrés sont donc indépendants des mesures de proximité mais fonction des rangs. En reprenant la représentation graphique précédente, nous avons matérialisé ces degrés par des traits reliant l'individu de référence à tous les autres, et dont le niveau de gris traduit l'amplitude. Plus le trait est foncé plus le degré est élevé (deux exemples sur deux individus de l'échantillon sont présentés sur la figure 4.5).

Les degrés peuvent donc être vus comme des scores (ou des utilités [169]) attribués en fonction des classements d'opinions. Puisqu'un individu induit un classement de tous les autres, il est lui même classé par chacun d'entre eux. Chaque membre de l'échantillon est donc caractérisé par les scores attribués individuellement par les autres, relativement aux opinions. Cette caractérisation nous amène alors à introduire une nouvelle notion.

Cette nouvelle notion pourrait être appelée la représentativité. Le but est de quantifier la façon avec laquelle un individu est préféré par l'ensemble de l'échantillon. On dira qu'un individu est très représentatif, au sein de l'échantillon, s'il est suffisamment préféré et par un nombre d'individus suffisamment grand. La représentativité d'un individu se déduit donc de façon naturelle de son classement auprès de chacun des autres. On quantifie donc la représentativité par agrégation des scores obtenus. On peut voir la représentativité comme la quantification d'une préférence collective à partir de préférences individuelles.
On représente graphiquement, sur notre exemple, la représentativité des individus au sein de l'échantillon, par des cercles autour des individus. Le gris est d'autant plus foncé que la représentativité d'un individu est importante (voir figure 4.6).

Ces deux notions permettent de représenter les individus vis à vis des autres, et vis à vis de l'échantillon pris dans sa globalité. Nous allons maintenant introduire une troisième notion caractérisant les liens entre les individus. Elle tente de répondre à la question : pour un individu donné, quels sont les membres de l'ensemble qu'il préfère et qui sont assez représentatifs ? Le concept sous-jacent est celui d'ensemble de connexion. On peut reformuler la question en : quels sont les individus qui le connectent à l'échantillon ? La réponse à cette question est suggérée par la première formulation. Cette troisième notion est en fait le résultat de la conjonction des deux premières. La connexion d'un individu a par conséquent deux propriétés évidentes :

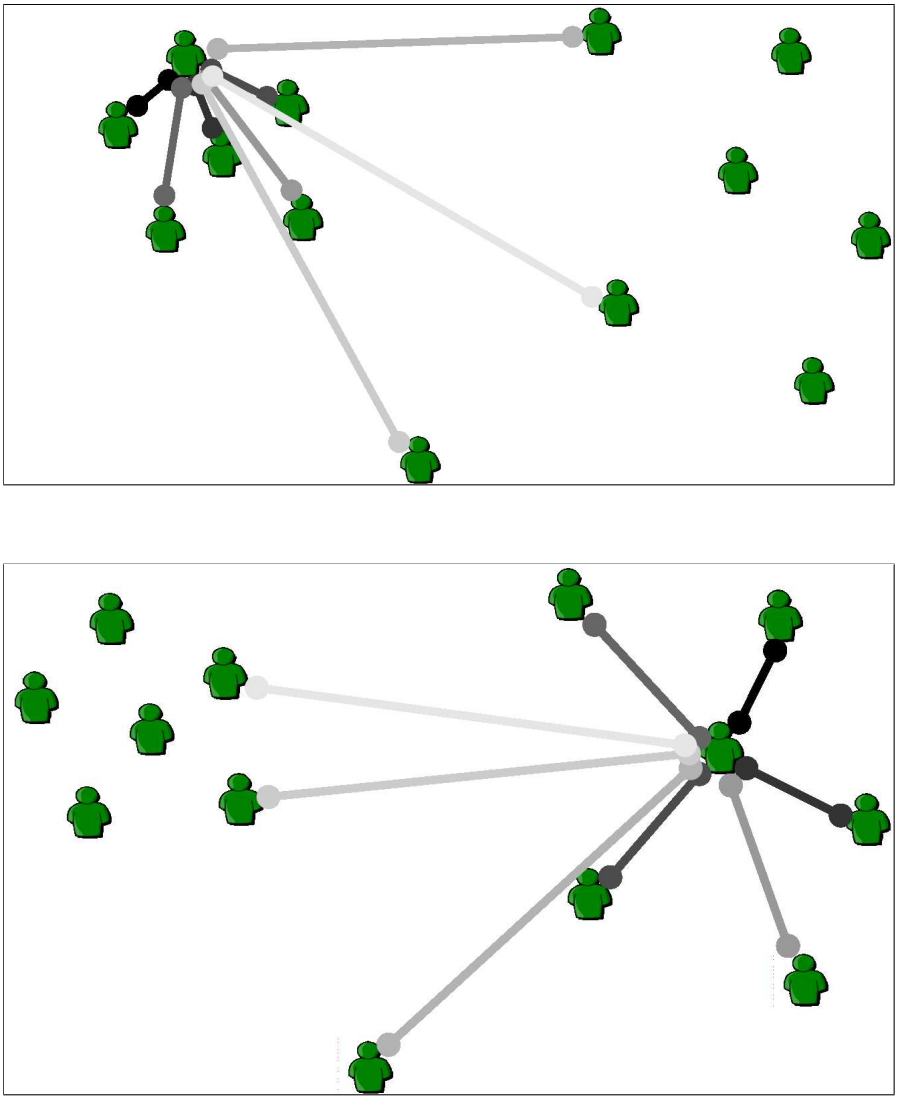


FIG. 4.5 – Exemple introductif : Chaque individu définit ses préférences sur l'ensemble des autres individus. On fait alors correspondre un degré à la position dans le classement obtenu. Graphiquement, on représente ce degré par un niveau de gris. L'individu dont les opinions lui sont le plus proches lui est relié par un trait noir, le second par un trait gris très foncé, le troisième par trait d'un gris un peu plus clair, et ainsi de suite.

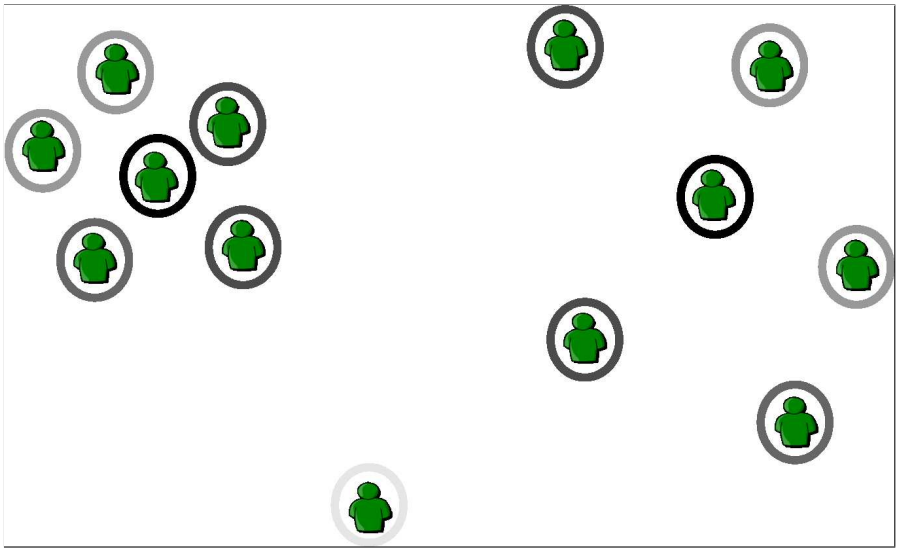


FIG. 4.6 – Exemple introductif : représentativité d'un individu dans l'échantillon. Le niveau de gris du cercle entourant un individu indique son degré de représentativité.

On considère un individu quelconque de l'échantillon.

- parmi deux individus d'égale représentativité, il se connectera à l'échantillon par celui qu'il préfère
- parmi deux individus qu'il "préfère autant", il se connectera à l'échantillon par celui qui est le plus représentatif

Ces deux propriétés sont illustrées sur les figures 4.7 et 4.8. La flèche rouge est dirigée du connecté vers le connecteur.

Cette notion de connexion nous permet alors directement de définir les groupement d'individus à effectuer. Il suffit en effet de regrouper les membres de l'ensemble initial en fonction de ces connexions.

Cet exemple nous a permis d'introduire de façon assez intuitive les notions sur lesquelles reposent notre système de représentation de données exposé dans la section suivante.

4.3.2 Représentation floue des données

Considérons un échantillon $X = \{x_1, x_2, \ldots, x_n\}$ de n données dans $\mathbb{R}^p$. On suppose que l'on dispose d'un tableau de dissimilarité ou de distance entre les individus, noté $D = (\delta_{i,j})_{i=1..n,j=1..n}$ (où $\delta_{i,j}$ est la dissimilarité entre les deux points x_i et x_j de X).

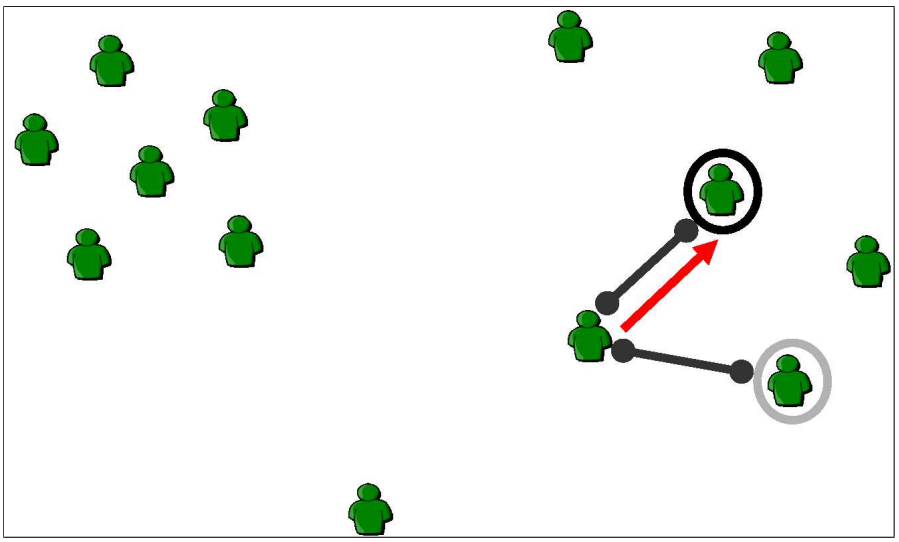


FIG. 4.7 – Exemple introductif : "parmi deux individus qu'il "préfère autant", il se connectera à l'échantillon par celui qui est le plus représentatif"

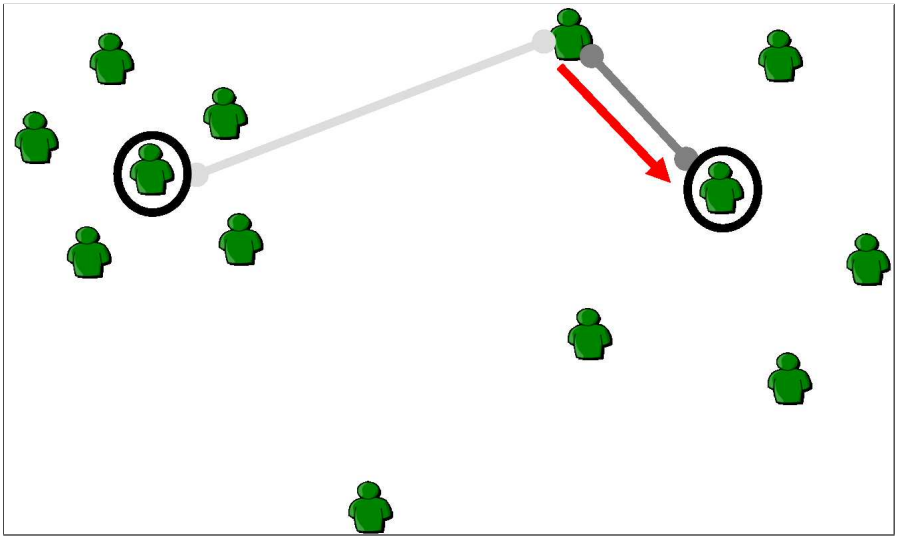


FIG. 4.8 – Exemple introductif : "parmi deux individus d'égale représentativité, il se connectera à l'échantillon par celui qu'il préfère"

4.3.2.1 Passage aux rangs

La première étape de notre méthode consiste donc à effectuer une transformation par rangs de ce tableau de dissimilarité. Autrement dit pour chaque individu, on trie les dissimilarités par ordre croissant. Chaque donnée induit donc un préordre sur l'échantillon (i.e. une relation binaire réflexive et transitive, sur X). On note $(\precsim_i)_{i=1..n}$ la famille de préordres correspondants, telle que :

$$\forall i \in [1,n], \forall j,k \in [1,n] : \quad x_j \precsim_i x_k \Leftrightarrow \delta_{i,j} \leq \delta_{i,k} \tag{4.6}$$

On peut également définir, pour tout individu une fonction R_i qui associe à un élément $x_j \in X$ son rang dans l'ensemble X ordonné par $\precsim_i$:

$$\forall i \in [1..N], R_i(x_j) = \sigma_i^{-1}(j) \tag{4.7}$$

où $(\sigma_i)_{i=1..n}$ est la famille de permutations sur $[1,n]$ telle que :

$$\forall i \in [1,n], x_{\sigma_i(1)} \precsim x_{\sigma_i(2)} \precsim \ldots \precsim x_{\sigma_i(n)}$$

Cette étape de transformation par rang est importante. On ne travaillera pas par la suite sur des distances mais sur des rangs. Cette approche va apporter de la robustesse aux processus utilisés ensuite.

De manière générale, en statistique, le passage aux rangs [60] permet de transformer des méthodes classiques en méthodes non paramétriques. Les hypothèses faites a priori sur les mesures sont plus faibles : la loi des distances est maintenant non-paramétrique. De plus, les représentations fournies sont robustes, très peu sensibles aux valeurs aberrantes.

Notre tableau de dissimilarité est maintenant transformé en tableau de rangs. Nous allons maintenant travailler avec ce tableau et définir les données comme sous-ensembles flous particuliers.

4.3.2.2 Les données comme sous-ensembles flous

Nous allons maintenant définir chaque donnée de l'échantillon comme un sous-ensemble flou [205]. Ce choix de représentation est opportun et motivé par la volonté de modéliser l'imprécision associée aux données particulières que sont les pixels d'une image multi-composante. La particularité de ces ensembles flous est que la fonction d'appartenance qui les caractérise est une fonction du rang des autres données et pas de leur valeur. Cette caractéristique fait donc que les profils des fonctions d'appartenances des données

sont tous identiques. On définit donc chaque point x_i comme un sous ensemble flou de X par les degrés d'appartenance :

$$\forall i \in [1,n], \forall j \in [1,n], \mu_{x_i}(x_j) = g(R_i(x_j)) \tag{4.8}$$

où g est une fonction discrète décroissante définie sur $[1,n]$ et telle que $0 \leq g \leq 1$. On peut par exemple choisir g comme une fonction Gaussienne :

$$\forall x \in [1,n], g(x) = e^{-\frac{(x-1)^2}{s^2}} \tag{4.9}$$

où s correspond donc en quelque sort à un paramètre de lissage.
Cette fonction g peut également être vue comme une fonction de score sur les rangs. Elle attribue un score à chaque position d'un classement. La figure 4.9 représente la fonction g comme définie par la relation 4.9 avec $s = 40$. L'axe des abscisses représente donc les rangs.

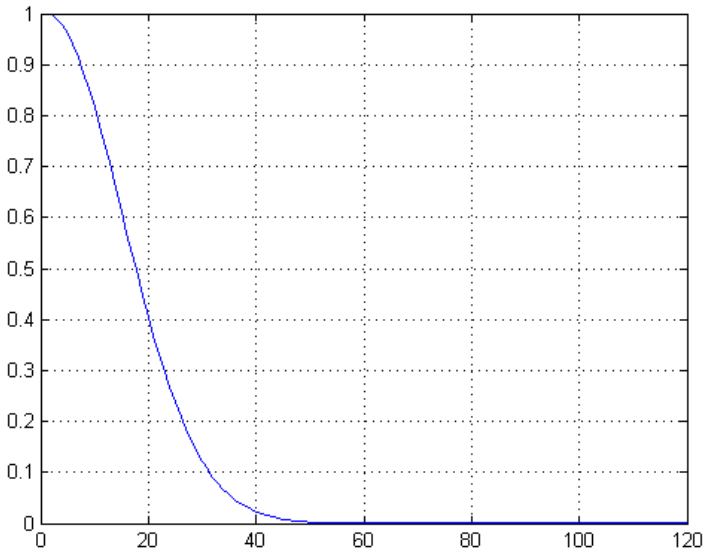


FIG. 4.9 – Degré d'appartenance des données en fonction de leur rang : cas d'une fonction g Gaussienne avec $s = 40$

Nous allons illustrer cette idée à l'aide d'un exemple simple. Considérons l'échantillon de données représenté sur la figure 4.10. Ces données sont simulées à l'aide de deux distributions Gaussiennes de 60 points chacune $(\mathcal{N}(5,2), \mathcal{N}(60,15))$.
Nous pouvons tout d'abord représenter le degré d'appartenance à une donnée arbitraire, en fonction des valeurs des autres données (et plus en fonction des rangs). La figure 4.11

(haut) représente les degrés d'appartenance à la donnée $x_1 = 9.1$ en fonction des valeurs des autres données. La figure 4.11 (bas) représente les degrés d'appartenance à la donnée $x_2 = 56.6.7057$. Ces deux figures représentent donc les fonctions d'appartenance de deux données en tant que sous-ensembles flous. Ces deux fonctions sont d'allure très différentes puisque les voisinages des deux données ne sont pas distribués de la même manière compte tenu de la construction de l'échantillon. Ces deux données appartiennent à l'une et l'autre des deux distributions. Il en résulte que les plus proches voisins de la première donnée sont contenus dans un intervalle plus petit que l'intervalle contenant les plus proches voisins de la seconde. Ce point illustre le caractère adaptatif de la représentation dû à l'utilisation des rangs plutôt que des distances. Par exemple, la distance d'un point à son k-ème voisin est plus grande dans les zones contenant peu de points que dans les régions en contenant beaucoup. Cet élément explique donc comment notre méthode permet de prendre en considération des groupes de données de densités très différentes.

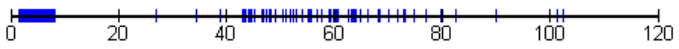


FIG. 4.10 – Echantillon (en dimension 1) de 120 données simulées par deux distributions Gaussiennes $\mathcal{N}(5,2), \mathcal{N}(60,15)$

4.3.2.3 L'échantillon comme sous-ensemble flou

L'échantillon à son tour va maintenant être représenté comme un ensemble flou. Pour définir les degrés d'appartenance à ce nouvel ensemble flou, nous utilisons une procédure d'agrégation des scores attribués à chaque donnée par toutes les autres. Cette étape peut donc être considérée comme une procédure d'agrégation. Les opérateurs d'agrégation sont nombreux et couramment utilisés pour agréger des sous-ensembles flous en logique floue, pour agréger des préférences en théorie de la décision ou des votes en théorie du choix social [61][77][171].

Nous avons choisi d'utiliser un opérateur de moyenne pondérée ordonnée ou OWA (Ordered Weighted Average). Ce choix s'avère le plus adapté à notre situation, il sera discuté dans la section de discussion 4.5. L'opérateur OWA est un cas particulier de l'intégrale de Choquet (intégrale floue) et est défini comme suit :

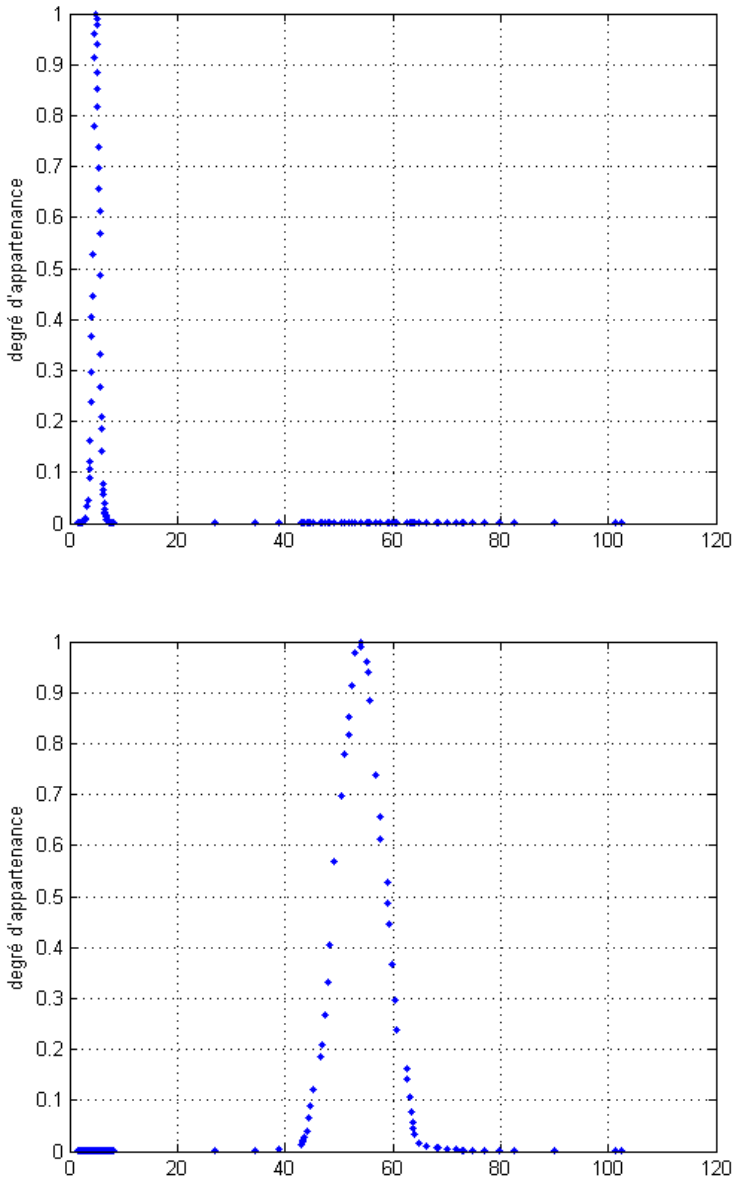


FIG. 4.11 – Fonctions d'appartenance des 30ème et 90ème données de l'échantillon

$$F_w : \begin{array}{l} \mathbb{R}^d \to \mathbb{R} \\ (y_1, y_2, \ldots, y_d) \mapsto \sum_{i=1}^{d} w_i.y'_i \end{array}$$

où $(y'_1, y'_2, \ldots, y'_d)$ est le vecteur des y_i triés par ordre croissant et w un vecteur de poids tels que $\sum_{i=1}^{d} w_i = 1$.

On utilise donc un tel opérateur pour agréger les sous ensembles-flous définis avant. On définit donc X comme un ensemble flou par :

$$\forall i \in [1, n], \mu_X(x_i) = F_w(\mu_{x_1}(x_i), \mu_{x_2}(x_i), \ldots, \mu_{x_n}) \tag{4.10}$$

Le choix des poids est ici un élément important (voir la section Discussion). En choisissant des poids égaux, l'opérateur OWA est équivalent à la moyenne classique. Nous avons choisi un manière particulière de répartir les poids. La figure 4.14 représente le profil de la répartition des poids que nous utilisons. Les premiers poids, w_i pour $i \in I_3$, sont tout d'abord fixés à zéro. Ceci permet de diminuer la contribution des quelques points les plus proches et donc de minimiser la contribution du bruit et des points (ou petits groupes de points) aberrants. Ensuite on maximise l'influence des plus proches voisins. Le paramètre I_2 permet de définir les contributions maximales. Enfin on considère qu'à partir d'un certain seuil (défini avec I_1), l'influence des données n'est plus significative, les poids sont fixés à 0.

On peut voir sur la figure 4.12 une représentation de la fonction d'appartenance à l'échantillon de la figure 4.10.

La notion que nous venons de définir est proche de celle de densité [13]. La représentation en 3 dimensions de la fonction d'appartenance à des similarités avec ce que serait l'estimation de la fonction de densité de probabilité. Notre fonction d'appartenance présente le même avantage de ne pas faire d'hypothèse sur la forme des classes comme le montre l'exemple en deux dimensions de la section suivante. Nous pouvons établir quelques éléments de comparaison entre les deux fonctions [13]. Notre fonction possède plusieurs caractéristiques intéressantes :

- elle permet de mettre en évidence avec la même importance deux classes de données ayant des densités différentes et des effectifs égaux, ce qui est assez difficile avec l'estimation de densité, le paramètre de lissage posant problème dans ce cas.
- les différences d'effectifs des classes n'altèrent pas leur représentation par notre fonction, contrairement à la densité estimée, pour les mêmes raisons.
- comme l'estimation de densité de probabilité, elle ne fait d'hypothèse sur la forme des classes.

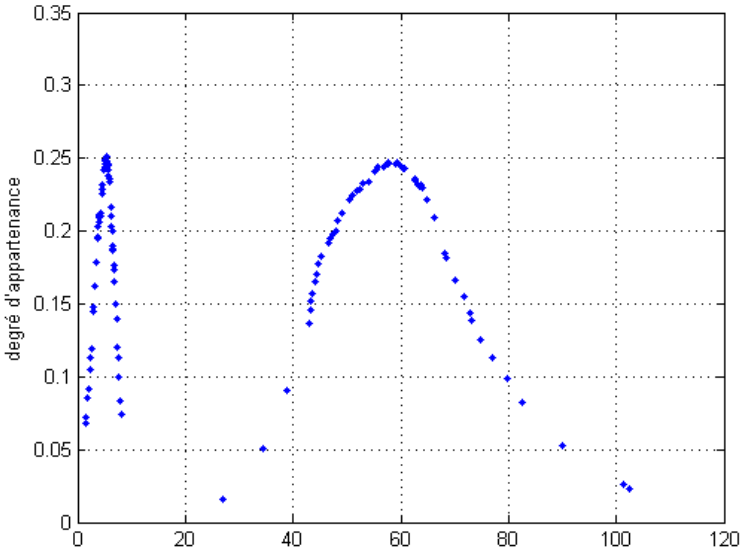


FIG. 4.12 – Fonction d'appartenance à l'échantillon de 120 données

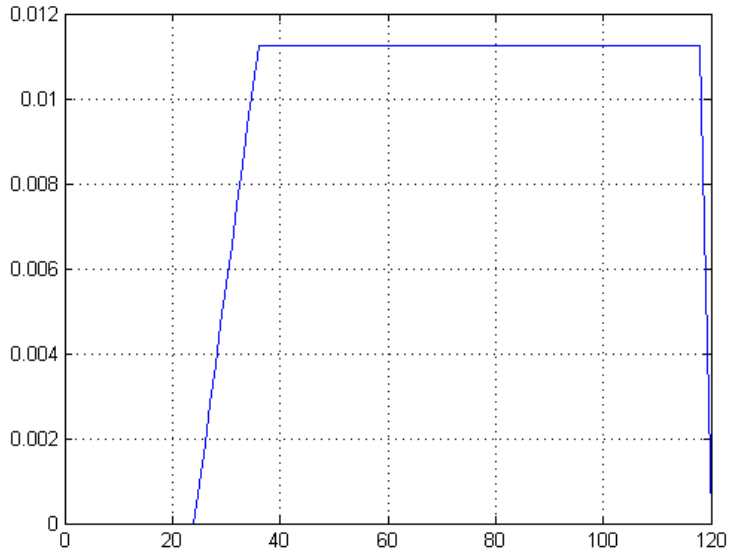


FIG. 4.13 – Poids utilisés pour représenter l'échantillon de 120 données

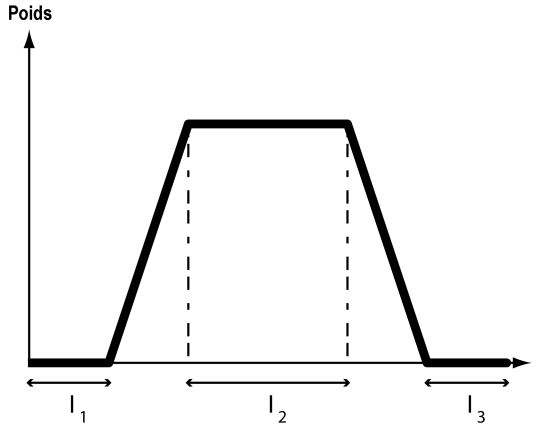


FIG. 4.14 – Construction des poids pour la procédure d'agrégation OWA

Lien avec le problème de classification. On pourrait s'arrêter ici dans la représentation et utiliser dès maintenant un algorithme de classification utilisant cette fonction comme fonction objectif. En effet, comme avec l'estimation de la densité de probabilité, on pourrait définir les classes directement à partir des modes de la fonction. Cette approche a donné lieu à une communication [13] dans laquelle la représentation utilise une fonction g uniforme et ou l'opérateur d'agrégation utilisé est une moyenne simple. Cette utilisation donne déjà des résultats satisfaisants et encourageants. Nous avons poursuivi notre développement et conçu la notion structurante d'ensembles de connexion des données.

4.3.2.4 Ensembles flous de connexion

La notion d'ensemble de connexion permet de représenter des liaisons entre les données. Elle utilise les deux définitions précédentes. Le but est de représenter la façon avec laquelle est connectée une donnée à l'échantillon.

On définit l'ensemble de connexion C_{x_i} d'une donnée x_i de X comme un sous ensemble flou. Le degré d'appartenance à C_{x_i} d'un point x_j de X est élevé lorsque son appartenance à X (vu comme ensemble flou) est élevée et que son appartenance au sous ensemble flou x_i est grande. Autrement dit, un point x_j connecte d'autant plus un point x_i qu'il appartient à cette données floue ET qu'il appartient à l'échantillon. Cette notion d'ensemble de connexion correspond donc à la conjonction des deux précédentes. On utilise la t-norme min pour réaliser cette conjonction.

On définit donc :

$$\forall i \in [1,n], \forall j \in [1,n], \mu_{C_{x_i}}(x_j) = min(\mu_X(x_j), \mu_{x_i}(x_j)) \tag{4.11}$$

Notons au passage que le concept de connexion n'est pas symétrique. Ce n'est pas parce qu'une donnée en connecte fortement une autre à l'échantillon que l'inverse est vraie. On a donc, en général : $\mu_{C_{x_i}}(x_j) \neq \mu_{C_{x_j}}(x_i)$.

La figure 4.15 représente les degrés de connexions flous des données de l'échantillon pour les points x_1 et x_2. La figure de gauche représente les degrés avec lesquels les données de l'échantillon (dont les valeurs sont représentées en abscisse) connectent la donnée x_1 à l'échantillon. La figure de droite représente ces degrés pour la donnée x_2.

Nous avons défini, dans cette partie, trois nouvelles notions en utilisant la théorie des ensembles flous. Chaque donnée de l'échantillon a donc tout d'abord été définie comme un sous ensemble flou dont la fonction d'appartenance est une fonction des rangs des autres données. Nous avons ensuite agrégé ces sous ensembles flous pour définir le sous ensemble flou associé à l'échantillon en entier. Enfin nous avons présenté le concept d'ensemble de connexion d'une donnée. Ce concept est déterminé en fusionnant les deux informations apportées par les deux premières notions et constitue une étape de découverte de structure ou de liaison entre les données. Les deux premières définitions sont des concepts de représentation des données et la troisième un concept de structuration de ces données.

Afin de valider notre méthode, nous allons en présenter une utilisation pour la classification de données. L'algorithme que nous avons choisi est assez simple et n'utilise pas le concept de classification floue. Nous allons voir qu'il donne déjà de bons résultats qui laisse supposer de riches perspectives dans le cadre d'une méthodologie "complètement" floue, c'est à dire utilisant notre méthode de représentation floue, puis effectuant la classification de façon également floue.

4.4 Application à la classification

Nous allons maintenant voir comment nous avons choisi de procéder pour obtenir une classification à partir des notions que nous venons de définir.

4.4.1 Principe

La notion d'ensemble de connexion définie précédemment induit de façon naturelle une structure de graphe sur les données. Il existe deux possibilités pour construire ce

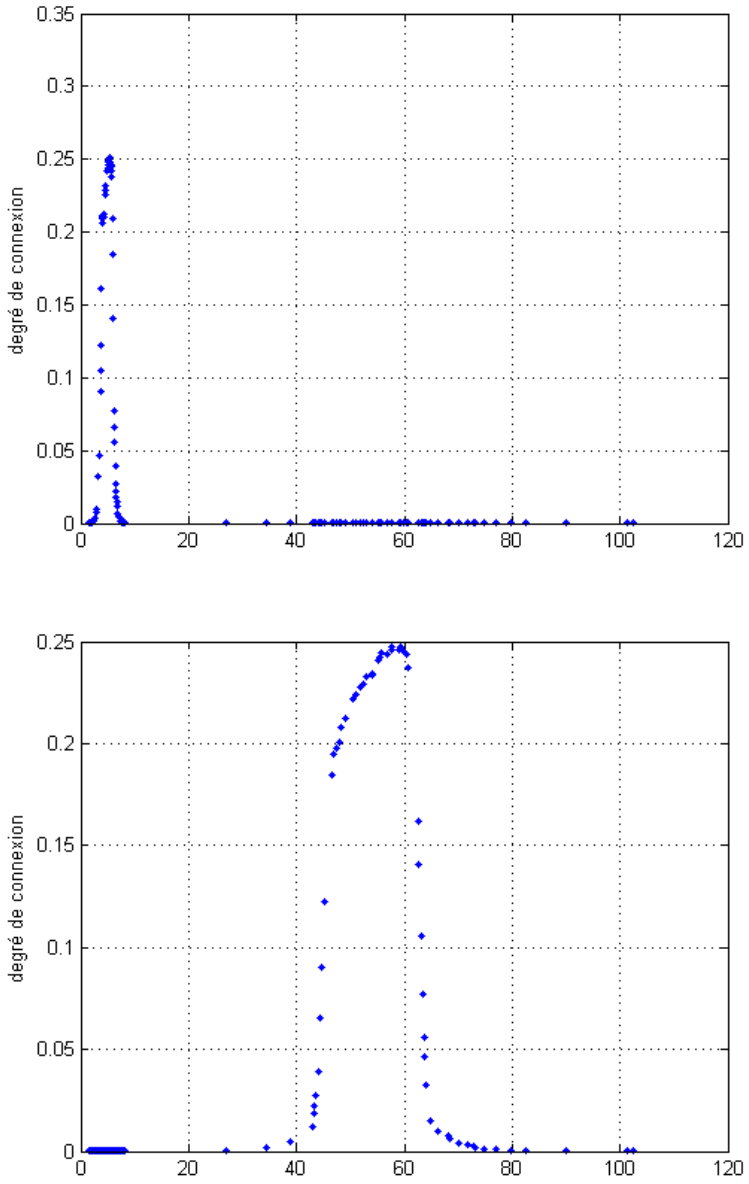


FIG. 4.15 – Connexion floue pour les 30ème et 90ème données de l'échantillon

graphe.

La première consiste à associer aux données le graphe complet dont les arcs sont valués par les degrés d'appartenance aux ensembles de connexion (c'est à dire les $\mu_{X_{x_i}}(x_j)$). On définit donc le graphe $G = (V, E)$ où V l'ensemble des sommets est constitué des individus x_i de l'échantillon X, et E l'ensemble des arcs comporte tous les arcs possibles entre deux sommets. Le poids $p_{i,j}$ de l'arc $(x_i \rightarrow x_j)$ est défini par :

$$p_{i,j} = \mu_{X_{x_i}}(x_j) \tag{4.12}$$

La deuxième façon, tout aussi intuitive, est une procédure de défuzzyfication des ensembles de connexions qui permet de construire le graphe $G = (V, E)$ dont l'ensemble des sommets V est constitué des individus de l'échantillon et E l'ensemble des arcs est tel que :

$$s = (x_i \rightarrow x_j) \in E \Leftrightarrow x_j \text{ connecte } x_i \text{ à } X \tag{4.13}$$

où l'on dit que x_j connecte x_i à l'échantillon s'il est le point dont le degré d'appartenance à C_{x_i} est le plus élevé. Autrement dit :

$$x_j \text{ connecte } x_i \text{ a } X \Leftrightarrow \underset{x_k \in X}{Argmax}(\mu_{C_{x_i}}(x_k)) \tag{4.14}$$

C'est cette approche de classification que nous avons retenue en première intention. Elle est très proche de l'algorithme DBSCAN et s'inscrit dans cette famille de méthodes comme DENCLUE ou d'autres.

Le processus de classification proprement dit est basé sur l'utilisation de la structure de graphe (voir par exemple les méthodes proposées dans [201][199][88][85][75] ou [180]). On définit les classes comme les composantes connexes du graphe obtenu. La détection des composantes connexes d'un graphe est un problème classique que l'on sait résoudre facilement. Un parcours "en profondeur d'abord" du graphe fournit un résultat avec une complexité en nombre d'opérations de $O(max(|V|, |E|))$. On rappelle qu'une composante connexe d'un graphe orienté est un sous-graphe (i.e. la restriction d'un graphe à un sous-ensemble de ses sommets) tel qu'il existe une chaîne (i.e. une succession quelconque d'arcs adjacents) entre tout couple des sommets qui le constituent.

Nous allons illustrer cette technique à l'aide de quelques exemples sur des données simulées (qui vont nous permettre d'illustrer les caractéristiques de notre méthode et de les valider), puis deux exemples d'application à la classification de données réelles seront présentés : l'un sur une image multicomposante et l'autre sur les données IRIS.

4.4.2 Exemples et applications

Cette technique est d'abord testée sur des données simulées. Son utilisation en conditions contrôlées va nous permettre de montrer son intérêt.

4.4.2.1 Données simulées

Premier exemple Nous présentons ici un premier exemple qui va nous permettre de constater et de confirmer le fait que notre méthode ne fait pas d'hypothèse sur la forme des classes.

Considérons donc un échantillon en deux dimensions, représenté sur la figure 4.16. Cet échantillon est constitué de deux distributions de 200 et 200 données simulées.

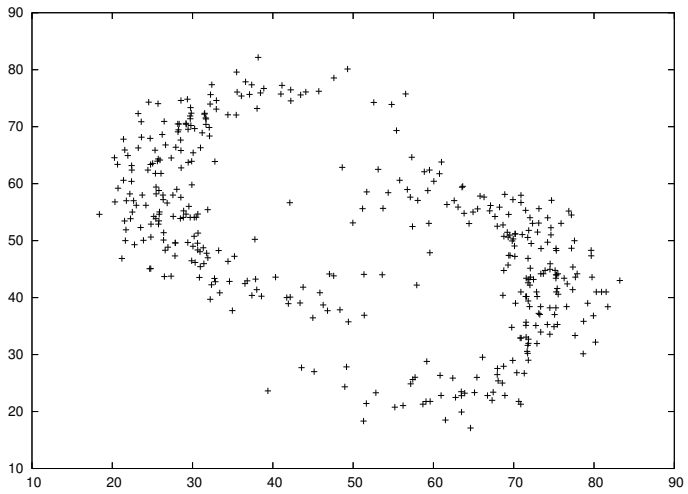


FIG. 4.16 – Exemple 1 : Echantillon simulé , en dimension 2, de 400 données

On construit maintenant les représentations floues décrites dans cette section. On utilise $s^2 = 65$ pour la fonction Gaussienne g pour la fonction d'appartenance aux données floues.

La figure 4.17 représente en 3 dimensions la fonction d'appartenance μ_X à l'échantillon X avec $I_1 = [0, 5], I_2 = [0, 350], I_3 = \emptyset$ pour la création des poids. Les deux premières dimensions correspondent aux valeurs des données et μ_X est représentée sur la troisième dimension.

Après avoir déterminé les ensembles de connexion flous, on construit le graphe comme indiqué avant (Figure 4.18).

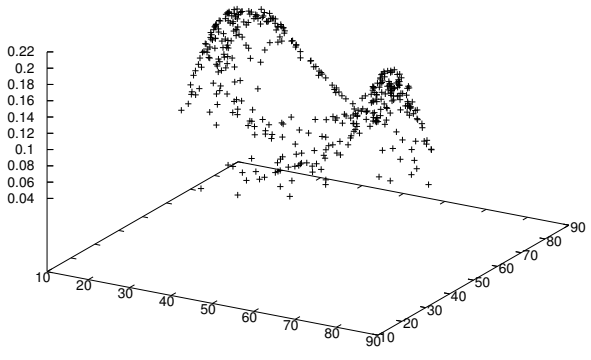


FIG. 4.17 – Exemple 1 : Fonction d'appartenance à l'échantillon de 400 données

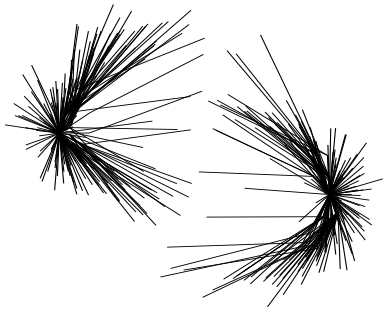


FIG. 4.18 – Exemple 1 : Graphe obtenu à partir de la défuzzyfication des ensembles de connexion de l'échantillon de 400 données

Les classes sont alors déterminées par les composantes connexes du graphe. On observe que la séparation non linéaire entre les classes est bien détectée (Figure 4.18). Cet exemple nous confirme expérimentalement le fait que notre méthode ne fait pas d'a priori sur la forme des classes. Sur ce type de données, un algorithme comme les k-means, qui privilégie les classes de forme sphérique ou elliptique, ne parviendrait à affecter les extrémités des distributions en forme de "croissants" aux bonnes classes.

Deuxième exemple Voici donc maintenant le second exemple. Cette fois, 3 classes de données, en dimension 2, sont simulées selon trois distributions Gaussiennes $\mathcal{N}(0, 8)$, $\mathcal{N}(15, 2)$ et $\mathcal{N}(25, 2)$. Ces trois classes ont des effectifs respectifs de 200, 100 et 100 données. L'échantillon ainsi constitué est représenté sur la figure 4.19. La figure 4.20 représente la fonction d'appartenance à l'échantillon flou calculée pour chaque point des données. Nous avons utilisé une fonction g Gaussiennes avec $\sigma =$. Le vecteur des poids, représenté sur la figure 4.22. A titre de comparaison, nous avons calculé l'estimation de la fonction de densité de probabilité par la méthode de Parzen. On observe que notre fonction est beaucoup plus informative que la densité estimée sur cet exemple.
Enfin, la figure 4.23 représente le graphe obtenu à partir des ensembles de connexion et qui détermine immédiatement les classes par association avec les composantes connexes. Les composantes connexes correspondent exactement aux "vraies" classes qui composent l'échantillon. Cet exemple simple nous montre que notre méthode fonctionne bien avec des classes de densités et d'effectifs différents.

Troisième exemple Dans cet autre exemple, 3 classes de données, en dimension 2, sont simulées selon trois distributions Gaussiennes $\mathcal{N}(0, 4)$, $\mathcal{N}(4, 16)$ et $\mathcal{N}(4, 28)$. Les effectifs respectifs des classes sont de 500, 200 et 50 données. Les figures 4.24, 4.25 et 4.28 représentent respectivement l'échantillon, la fonction d'appartenance à l'échantillon évaluée en chaque point des données, et le graphe résultat. Ces résultats ont été obtenus avec un paramètre $\sigma = 50$ et un vecteur poids représenté surla figure 4.27.

Ces trois exemples d'utilisation sur des données simulées nous ont permis de vérifier les caractéristiques de notre méthode. Nous allons maintenant proposer deux exemples d'application à des données réelles. La première application est effectuée sur une image multicomposante.

4.4.2.2 Image multicomposante

On considère l'image multicomposante que nous avons déjà vue et présentée sur la figure 4.29

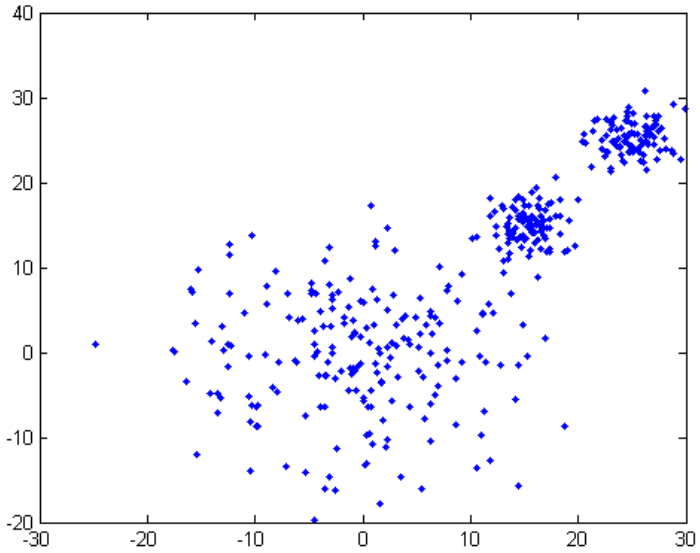


FIG. 4.19 – Exemple 2 : Echantillon, en dimension 2, composé de 3 classes de 200, 100 et 100 données

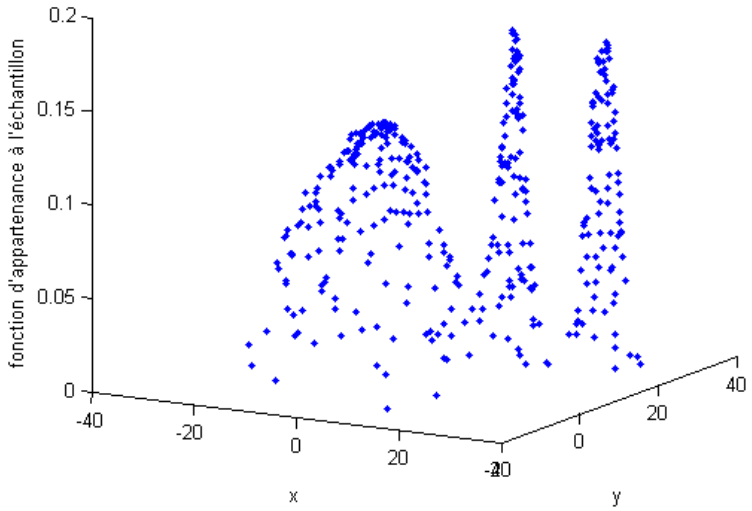


FIG. 4.20 – Exemple 2 : Fonction d'appartenance à l'échantillon. Cette fonction est moins sensible aux problèmes liés à une densité faible

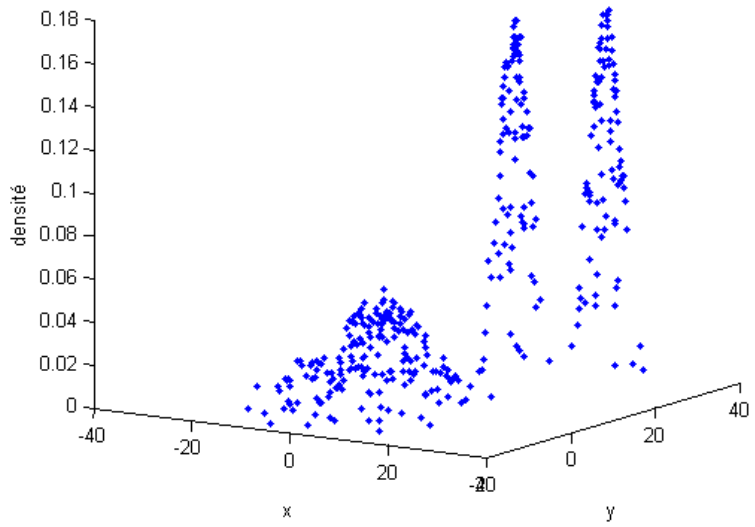


FIG. 4.21 – Exemple 2 : Estimation de la fonction de densité de probabilité de l'échantillon par la méthode de Parzen. Dans les queues de distribution ou lorsque la densité est faible, cette fonction est très bruitée

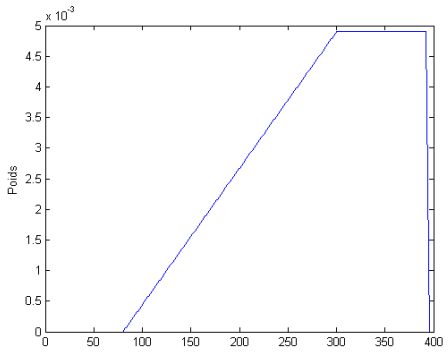


FIG. 4.22 – Exemple 2 : Vecteur de poids utilisé pour l'opérateur d'agrégation

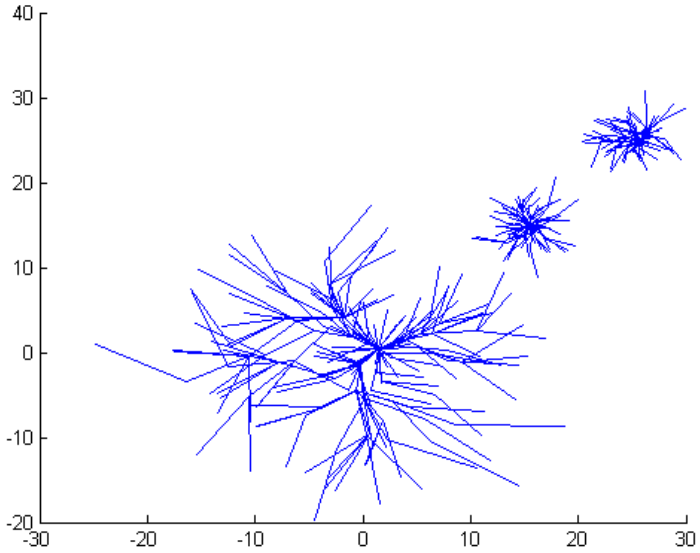


FIG. 4.23 – Exemple 2 : Graphe associé à l'échantillon, construit à l'aide des ensembles de connexion et dont les composantes connexes déterminent les classes de données

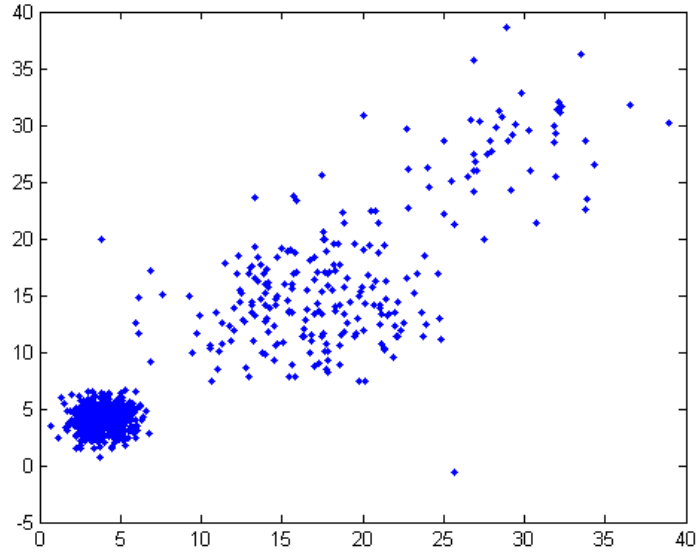


FIG. 4.24 – Exemple 3 : Echantillon, en dimension 2, composé de 3 classes de 500, 200 et 50 données

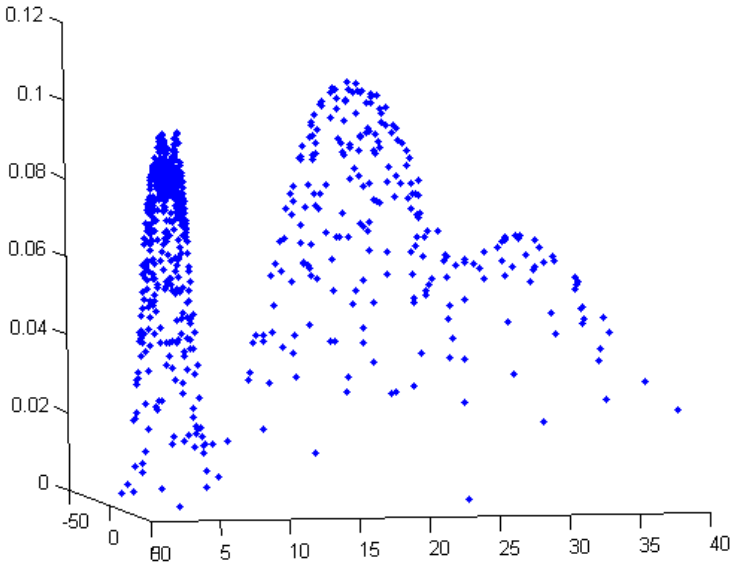


FIG. 4.25 – Exemple 3 : Fonction d'appartenance à l'échantillon. Cette fonction est moins sensible aux variations de densités et d'effectifs que la densité de probabilité (voir figure 4.26), et permet de révéler les trois classes de l'échantillon

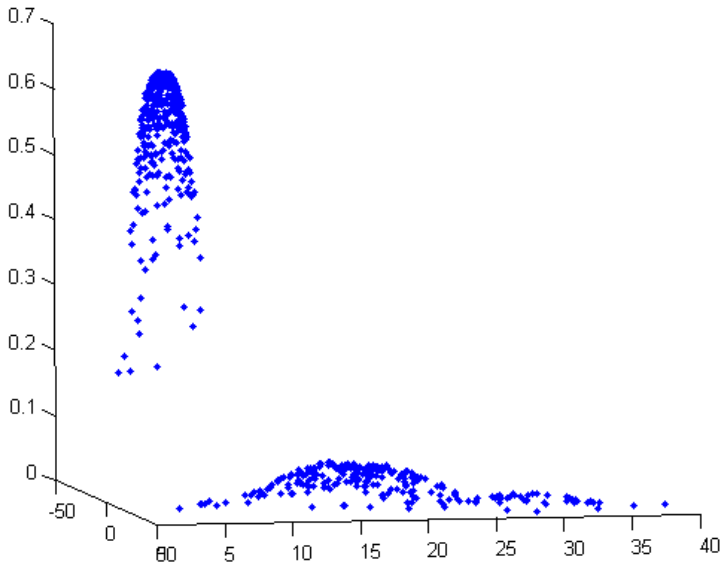


FIG. 4.26 – Exemple 3 : Estimation de la fonction de densité de probabilité de l'échantillon par la méthode de Parzen. Une seule classe est bien révélée par cette fonction qui est sensible aux variations de densités et d'effectifs

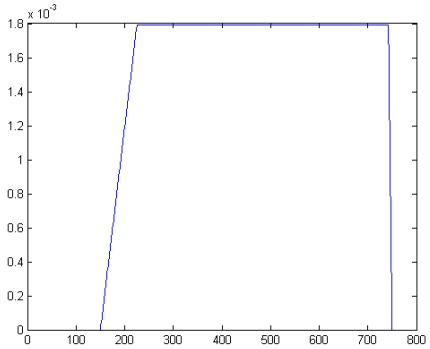


FIG. 4.27 – Exemple 3 : Vecteur de poids utilisé pour l'opérateur d'agrégation

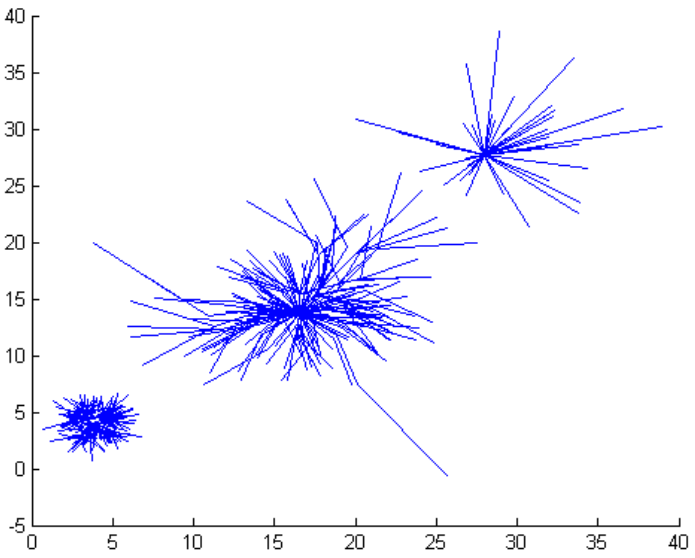


FIG. 4.28 – Exemple 3 : Graphe associé à l'échantillon, construit à l'aide des ensembles de connexion et dont les composantes connexes déterminent les classes de données

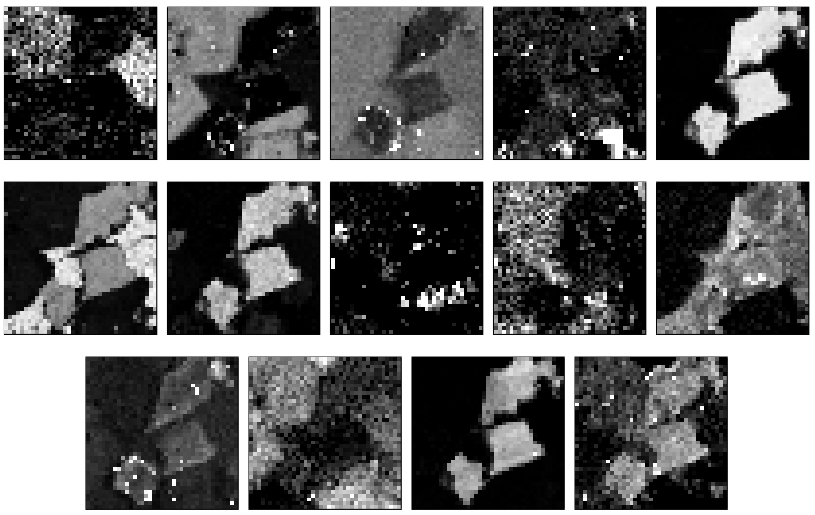


FIG. 4.29 – Image à 14 composantes

Le résultat fourni par notre méthode est présenté sur la figure 4.30. Quatre classes de données ont été déterminées. Elles sont représentées par quatre niveaux de gris arbitraires différents sur cette image. Il peut être intéressant de comparer ce résultat avec celui obtenu [96] avec la méthode de M. Herbin, N. Bonnet et P. Vautrot (voir figure 4.31). Le nombre de classes est identique et on sait qu'il existe 4 vraies classes. On note quelques petites différences sur quelques points (notamment sur les frontières des classes), mais les structures détectées sont globalement similaires.

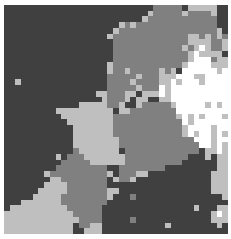


FIG. 4.30 – Résultat de la classification des pixels de l'image multicomposante de Fluorescence X. Chacune des 3 classes de pixels est représentée par un niveau de gris arbitraire différent

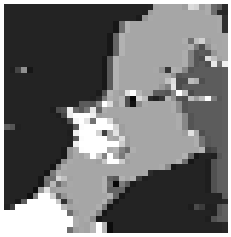


FIG. 4.31 – Résultat obtenu avec la méthode *Herbin et al*

Cette première application montre donc l'intérêt de notre méthode pour la classification d'images multicomposantes. Voyons maintenant une utilisation sur des données quelconques, les données IRIS.

4.4.2.3 Données réelles

Nous avons appliqué cet algorithme aux données de la base IRIS [150][12]. Le résultat est assez satisfaisant. En effet nous obtenons 91.3% de bon classement. Les paramètres ayant fourni cette classification sont les suivants :

- $\sigma = 15$
- $I_1 = [0, 52]$

- $I_2 = [90, 127]$
- $I_3 = \emptyset$

Rappelons que cette base est constituée de 3 classes de 50 données chacune.
A titre de comparaison, nous avons testé l'algorithme des k-means qui nous a fourni des résultats variant entre 69% et 90% de bons classements en fonction de l'intialisation (l'algorithme utilisait un sytème de préclassification sur un sous-échantillon pour déterminer les centres initiaux).

4.5 Discussion et conclusion

Nous avons présenté, dans la section précédente, notre nouvelle approche de représentation des données. Nous allons maintenant discuter certains points abordés dans l'exposé "technique".

Redondance d'information Le premier point -anecdotique- que nous allons discuter est celui de la redondance de la manipulation des informations. En effet d'un point de vue strictement algorithmique, le processus constitué de plusieurs étapes, que nous proposons d'appliquer aux données, modifie les données par application successive de transformations (sur les rangs) qui pourraient effectivement être regroupées en une seule opération.
Il faut cependant rappeler que l'élaboration de notre méthode était également motivée par un soucis d'interprétabilité. L'interprétation a d'ailleurs une grande importance dans les problèmes de classification : en effet si la conception d'un algorithme s'arrête à l'étape de labélisation, il s'agit ensuite de donner du sens aux regroupements fournis. Les étapes de notre représentation utilisent la théorie des sous-ensembles flous qui permettent de donner de la signification aux notions introduites.

Un autre point de vue Nous avons donc utilisé le formalisme flou pour définir cette nouvelle façon de représenter les données. En effet puisque les données que nous sommes amenés à traiter sont des pixels, la théorie des ensembles flous est tout à fait adaptée au caractère incertain et imprécis des informations.
Pourtant, sans affecter les opérations effectuées sur les données, il est possible de voir les choses avec un autre point de vue. Cette autre façon de voir les choses est proche de la façon dont ont été présentées les notions dans l'exemple introductif. On peut considérer en effet l'échantillon comme un ensemble d'individus qui expriment leurs préférences sur l'ensemble de ce même échantillon. Un score est alors attribué aux positions obte-

nues, puis ces informations sont agrégées (à l'aide d'un opérateur OWA) pour déterminer un choix collectif sous forme de score global. Ce point de vue ramène les notions à un problème classique en théorie du choix social ou en agrégation multicritère [63][79]. Le problème classique sous-jacent est généralement appelé *agrégation de rangs*.
La différence -et c'est ici un avantage- entre notre approche et ces problèmes classiques est que nous nous contentons d'affecter un score (via le calcul du degré d'appartenance floue à l'échantillon), nous ne cherchons pas à déterminer un ordre global optimal, comme en théorie des votes. Nous ne nous heurtons donc pas aux problèmes rencontrés lors de la recherche de procédures de vote optimales. Ces similarités nous incitent cependant à explorer les travaux réalisés dans ces disciplines pour affiner ou adapter notre modèle.

Choix des paramètres Toutes les méthodes de classification utilisent des paramètres : un nombre de classes maximum pour les k-means, le paramètre de lissage pour les méthodes basées sur la densité, etc...
Le premier paramètre de notre technique de représentation est le σ de la fonction Gaussienne g, pour le calcul des fonctions d'appartenance aux données floues. Ce paramètre a un rôle similaire à celui de la taille de la fenêtre de Parzen dans les méthodes de densité. C'est donc notre paramètre de lissage. L'estimation automatique de ce paramètre est un problème difficile. Mais les solutions proposées dans les autres approches sont applicables à notre cas. La solution la plus simple consiste à réutiliser la méthode adoptée dans l'algorithme DBSCAN et rester ainsi dans cette "famille" de méthodes. Une autre solution est de reprendre la méthode développée pas M. Herbin dans [96] qui consiste à utiliser une méthode de bootstrap [66] pour estimer ce paramètre.
On peut enfin considérer que les poids dans la procédure d'agrégation sont aussi un (plusieurs) paramètre. Nos recommandations sur le choix des poids sont de nature expérimentale, à l'instar de la façon dont est déterminé le paramètre $MinPts$ dans DBSCAN [67]. Une très petite amplitude pour l'intervalle I_1 suffit généralement, tandis qu'une grande taille pour I_2 (relativement au nombre de données) donne de bons résultats. Mais ce choix a une interprétation liée à la nature de l'opérateur OWA et dont nous parlerons dans le paragraphe suivant.
Nous avons constaté lors de nos expériences et nos différents tests, que le paramètre le plus déterminant quant à l'allure la fonction d'appartenance floue à l'échantillon est le paramètre de lissage σ. Le choix des poids apparaît comme un raffinement supplémentaire lorsque les données sont bruitées ou en présence de points aberrants.

Choix de la méthode d'agrégation Nous avons choisi, pour agréger les sous-ensembles flous que constituent les données, d'utiliser une moyenne ordonnée pondérée (OWA)[202].

Il existe cependant un grand nombre d'opérateurs d'agrégation d'information différents [62]. L'approche que nous avons adoptée nous suggérait d'utiliser un OWA. En effet, cet opérateur d'agrégation est un compromis entre les opérateurs conjonctifs ("OU") et disjonctifs ("ET"). La somme pondérée ordonnée est un cas particulier d'intégrale floue [78] (concept introduit par Sugeno qui étend celui de Lebesgue). Dans un OWA, les poids ne portent plus sur les sources mais sur les rangs des quantités sommées. Des poids bien choisis ramènent l'OWA à des cas particuliers correspondant à des opérateurs classiques :

- lorsque $\omega_1 = 1$ (et $\omega_i = 0$ pour $i > 1$), on retrouve l'opérateur minimum ;
- $\omega_n = 1$: opérateur maximum ;
- si n est impair, $\omega_{\frac{n+1}{2}} = 1$: médiane (si n est paire, $\omega_{\frac{n}{2}} = \omega_{\frac{n}{2}+1} = \frac{1}{2}$;
- on retrouve la moyenne classique en prenant des poids égaux.

La façon dont on choisit les poids de l'OWA donne donc une tendance, un caractère, à l'opérateur d'agrégation correspondant.

Développements futurs ou engagés Tous les points évoqués dans cette section font l'objet ou vont faire l'objet de travaux engagés ou à venir. Des études plus poussées du comportement de représentation en changeant l'opérateur d'agrégation vont être menées. Dans l'immédiat, nos efforts vont se concentrer sur l'élaboration d'un algorithme de classification adapté à notre mode de représentation et dédié à la segmentation d'images multicomposantes.
Une autre piste de travail a été ouverte pour construire une méthode de réduction de dimensionnalité non linéaire basée sur notre représentation et sur la notion d'observateur.

Nous avons développé un nouveau type de représentation des données pour la classification. Le but de ce travail est de fournir une représentation basée sur le flou adaptée aux contraintes liées aux données particulières que sont les images. Nous avons constaté que les méthodes classiques de classification n'étaient pas nécessairement bien adaptées aux caractéristiques de ces données.

Notre contribution se situe en amont du processus de classification proprement dit. Nous avons en effet choisi d'améliorer la façon de considérer et de représenter les données avant de les classer. Notre méthode de représentation permet ensuite d'appliquer différents types d'algorithmes de classification. Cette méthodologie est composée d'une phase de représentation des données et de l'échantillon qu'elles constituent, puis d'une phase de construction de liaisons entre elles. Ces deux phases constituent deux étapes de représentation puis de structuration de l'ensemble de données.

L'apport du flou se traduit par deux aspects. Le premier réside dans la flexibilité et l'adaptation aux contraintes que son usage induit. Le second aspect est la facilité

d'interprétation qui en découle. Les ensembles flous ont en effet un pouvoir significatif important qui se révèle particulièrement adéquat dans une utilisation orientée image. En effet, comme nous l'avons vu en introduction de ce chapitre, les caractéristiques des moyens d'acquisition des images engendrent de l'*imprécision* sur les données qui se traduit donc par une *incertitude* lors de l'analyse et l'étude de ces données. Or ces deux caractères sont un terrain propice et dédié à l'utilisation de la théorie des ensembles flous.

Notre travail constitue donc un nouveau cadre pour la classification de données particulièrement adapté aux données images. Il a déjà été valorisé par deux communications [13][18] et représente une base pour des développements en cours ou futurs, en analyse d'images. Par ailleurs, si nous l'avons conçu dans un contexte et un but de traitement d'images, le caractère "pluridisciplinaire" permet d'imaginer des développements transverses. En effet l'utilisation de concepts empruntés à la décision multicritère, la théorie du choix social et les statistiques non-paramétriques en font une méthodologie utilisable pour des types de données très différents. Notre travail actuel se concentre actuellement sur le développement d'un algorithme de segmentation d'image utilisant ce concept.

Chapitre 5

Conclusion

Nous avons donc proposé, au cours de ce travail de thèse, deux contributions en analyse exploratoire de données. La première est une approche de visualisation des données, la seconde, une approche de classification. Ces deux démarches ont été motivées par le traitement d'images multicomposantes, mais ces contributions ont été développées dans un cadre plus général de données multidimensionnelles quelconques. Nous nous sommes également attachés, dans ce travail, à développer des méthodes non supervisées.

Notre première contribution se place dans le thème de la visualisation de données multidimensionnelles. La méthode que nous avons développée initialement pour les images multicomposantes a pu être généralisée aux données non spatialisées. Elle peut être considérée comme une technique orientée-pixels. Sa principale originalité est l'utilisation statistique de la couleur. L'attribution des couleurs s'effectue de manière automatique et complètement guidée par les données. Cette façon de procéder présente l'avantage de ne pas nécessiter d'apprentissage proprement dit. On n'introduit donc pas de subjectivité dans le processus de création. On obtient, à l'issue d'un processus rapide, une image synthétisant les principales informations contenues dans les données. L'image obtenue fournit une vision synthétique et immédiate des structures sous-jacentes. Dans le cas d'images multicomposantes, notre méthode permet d'exhiber les structures des pixels. Pour des données non spatialisées, elle arrange les données dans l'image finale en les regroupant par structures, sous forme de zones colorées connexes. Ces structures sont séparables visuellement grâce à la couleur. Notre méthode est donc un outil utile pour fournir une visualisation préliminaire avant une exploration plus fine.
Nous avons valorisé notre méthode par des publications dans différentes "communautés" : traitement du signal et des images [15][16][19], visualisation d'information [21], imagerie du vivant et médicale [17][14], ou fouille de données [18].

Nous avons développé une méthode statique de visualisation et un travail a été mené sur son adaptation et son utilisation dans un processus dynamique d'exploration. Le principe est d'utiliser notre visualisation avec différents points de références (le point de référence est le point selon lequel est calculée l'inertie du nuage de points). On obtient donc autant d'images différentes que nous avons choisi de points de référence. On définit alors une trajectoire, dans l'espace des données, sur laquelle on prend ces points et on obtient ainsi une séquence d'images de l'échantillon. Dans un futur proche, notre travail va consister à étudier la façon de définir ces trajectoires, et la vitesse de déplacement (c'est à dire la fréquence de création des plans de la séquence). Deux approches différentes sont envisagées. La première consiste à définir ces paramètres de déplacement de manière interactive et la seconde à les déterminer de façon non supervisée, complètement guidée par les données. Nous allons également nous intéresser aux singularités découvertes en observant l'échantillon. Les points singuliers correspondent à des points de rupture dans la séquence d'image, qui doivent nous permettre d'obtenir plus d'information sur les structures sous-jacentes des données.

Notre seconde contribution se situe en classification automatique de données. Plus exactement, elle se situe en amont du processus de classification. Nous avons proposé une nouvelle façon de représenter les données avant de les classer. Elle repose sur l'utilisation de la théorie des ensembles flous pour représenter les données comme des ensembles flous, et l'échantillon lui même comme un ensemble flou (et défini comme une agrégation des données floues). L'usage du flou permet de tenir compte du caractère ambigu de certains pixels et de l'imprécision qui en découle. Nous définissons également une notion de connexion (sous forme de sous-ensembles flous, à nouveau) qui permet de représenter les connexions implicites qui lient les données entre elles. Cette représentation a également une autre originalité. Les distances entre les données sont, au début de la méthode, transformées en rangs. Cette technique permet de donner une robustesse au concept. Nous avons pu valider l'intérêt de cette représentation en l'intégrant dans un processus complet de classification et en le testant sur divers exemples. Le premier algorithme que nous avons développé donne déjà de bons résultats. Pourtant cet algorithme n'utilise pas, dans l'étape de création des classes, le principe de la classification floue. Ce travail a été valorisé par deux communications [13][20].

Les perspectives de ce travail sont riches et variées. Le premier objectif des travaux à venir va être de développer un algorithme de classification floue spécialement dédié à la classification d'images multicomposantes (notamment pour des images de textures). L'utilisation de méthodes d'agrégation différentes ou plus générales pourrait également être testée. De manière générale, l'analogie entre notre représentation et des problèmes classiques d'autres domaines nous suggère d'explorer les méthodes propres à ces domaines

et à envisager de les tester dans notre contexte de la classification d'images multicomposantes.

Nous sommes partis, dans tout ce travail d'un problème de visualisation et de classification d'images multicomposantes. Les caractéristiques propres à ce type de données nous ont, de façon naturelle, incité à nous placer dans le contexte plus général de l'analyse de données multidimensionnelles. Nous avons ainsi développé des méthodes assez générales mais bien adaptées aux images. Cette généralisation nous a permis de sortir du cadre restrictif de l'image multicomposante et d'utiliser des techniques empruntées à des domaines de recherche variés, allant de la théorie du choix aux statistiques non paramétriques, ou de la décision dans l'incertain au traitement d'images.

BIBLIOGRAPHIE

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Proceedings of the 8th International Conference on Database Theory*. Springer-Verlag, 2001.

[2] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics : Ordering points to identify the clustering structure. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*, pages 49–60. ACM Press, 1999.

[3] D. Asimov. The grand tour : a tool for viewing multidimensional data. *Journal on Scientific and Statistical Computing*, 6(1) :128–143, 1985.

[4] A. De Backer, A. Naud, and P. Scheuders. Non linear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters*, 19 :711–720, 1998.

[5] D. M. Bates and D. G. Watts. *Nonlinear Regression Analysis and its Applications*. New York : Wiley, 1988.

[6] R. Bellman. *Adaptive control processes : a guide tour*. Princeton University Press, 1961.

[7] J.-P. Benzecri. Problèmes et méthodes de la taxonomie. Technical report, Rapport de recherche de l'Université de Rennes, France, 1965.

[8] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[9] J. Bertin. *Graphics and Graphic Information Processing*. Walter de Gruyter, Berlin, 1981.

[10] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[11] C. Bishop. *Neural Networks for Pattern Recognition*. Cambridge University Press, 1995.

[12] C.L. Blake and C.J. Merz. Uci repository of machine learning databases, 1998.

[13] F. Blanchard, H. Akdag, and M. Herbin. A new fuzzy representation for connecting data to a sample. In *MENDEL 2005, 11th International Conference on Soft Computing*, pages 114–119, Brno, Czech Republic, jun 2005.

[14] F. Blanchard and M. Herbin. Visualisation d'images multicomposantes par une image couleur. In *Imagerie pour les sciences du vivant et la médecine*, Strasbourg, sep 2003.

[15] F. Blanchard and M. Herbin. L'image couleur pour visualiser des données multidimensionnelles. *Traitement du Signal (TS)*, 21 :453–460, nov 2004.

[16] F. Blanchard and M. Herbin. Vimci : Visualization of multidimensional images through color image. In *The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Color Image Processing & Applications Special Session*, Orlando, Florida, USA, jul 2004.

[17] F. Blanchard and M. Herbin. *Visualisation d'Images Multicomposantes par une Image Couleur*, pages 303–308. M. Faupel, P. Smigielski, R. Grzymala (Fontis/Formatis), sep 2004.

[18] F. Blanchard, M. Herbin, and F. Rousseaux. Compendium de données multidimensionnelles par une image couleur. In *EGC 2005 Atelier Visualisation et extraction de connaissances*, Paris, jan 2005.

[19] F. Blanchard, M. Herbin, and P. Vautrot. Visualization of high dimensional data through a single color image. In *3rd IASTED International Conference on Visualisalization, Imaging and Image Processing (VIIP'03)*, pages 874–879, Benalmádena, Spain, sep 2003. M.H. Hamza.

[20] F. Blanchard, M. Herbin, and P. Vautrot. Vers une classification non supervisée basée sur un nouvel indice de connectivité (accepte). In *GRETSI 2005*, Louvain-la-Neuve, sep 2005.

[21] F. Blanchard, L. Lucas, and M. Herbin. A new pixel-oriented visualization technique through color image. *Information Visualization*, 5 :(ACCEPTE, A PARAITRE), 2005.

[22] N. Bonnet, M. Herbin, and P. Vautrot. Extension of the scatterplot approach to multiple images. *Ultramicroscopy*, 60 :349–355, 1995.

[23] B. Bouchon-Meunier. *La logique floue*. PUF, 1993.

[24] B. Bouchon-Meunier. *La logique floue et ses applications*. Addison-Wesley, Paris, 1995.

[25] N. Boujemaa. Generalized competitive clustering for image segmentation. In *In The 19th International Meeting of the North American Fuzzy Information Processing Society*, 2000.

[26] M. R. Brito, A. Quiroz, and J. E. Yukich. Graph-theoretic procedures for dimension identification. *Journal of Multivariate Analysis*, 81 :57–84, 2002.

[27] D. S. Broomhead and M.Kirby. A new approach to dimensionality reduction : Theory and algorithms. *SIAM Journal of Applied Mathematics*, 60(6) :2114–2142, 2000.

[28] J. Bruske and E. Merényi. Estimating the intrinsic dimensionality of hyperspectral images. In *ESANN'99, European Symposium on Artificial Neural Networks, Bruges (Belgium)*, pages 105–110, 1999.

[29] J. Bruske and G. Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5) :572–575, May 1998.

[30] M. Bruynooghe. Recent results in hierarchical clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(3) :541–571, 1993.

[31] A. Buja, D. Cook, and D. F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5 :78–99, 1996.

[32] A. R. Butz. Alternative algorithm for hilbert's space-filling curve. *IEEE Transactions on Computing*, C-20 :424–426, 1971.

[33] F. Camastra. Data dimensionality estimation methods : a survey. *Pattern Recognition*, 36(12) :2945–2954, 2003.

[34] F. Camastra and A. Vinciarelli. Estimating intrinsic dimension estimation of data with a fractal-based approach. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 24(10) :1404–1407, 2002.

[35] K.S. Card, J.D. Mackinlay, and B. Schneiderman. *Readings in Information Visualization, Using Vision to Think*. Morgan Kaufmann, California, USA, 1999.

[36] J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1) :157–192, 1999.

[37] J. F. Cardoso and P. Comon. Independent component analysis, a survey of some algebraic methods. In *Proc. ISCAS Conference*, volume 2, pages 93–96, 1996.

[38] M.A. Carreira-Perpinan. A review of dimension reduction techniques. Technical Report CS–96–09, Dept. of Computer Science, University of Sheffield, January 1997.

[39] G. Celeux, E. Diday, G. Covaert, Y. Lechevallier, and H. Ralambondrainy. *Classification automatique des données*. Dunod Informatique, Paris, 1989.

[40] C. Chen. *Information Visualization and Virtual Environments*. Springer, 1999.

[41] C. K. Chen and H. C. Andrews. Nonlinear intrinsic dimensionality computations. *IEEE Transactions on System Man and Cybernetics*, 3 :197–200, 1973.

[42] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8) :790–799, 1995.

[43] B.M. Collins. *Data visualization - has it all been seen before ?*, chapter 1, pages 3–28. Academic Press, London, 1993.

[44] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. on Pattern Analysis and MAchine Intelligence*, 25(2), 2003.

[45] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *IEEE Int. Conf. Computer Vision(ICCV'99)*, pages 1197–1203, Kerkyra, Greece, 1999.

[46] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis Machine Intelligence*, 24(5) :603–619, 2002.

[47] P. Comon. Independant component analysis, a new concept ? *Elsevier, Signal Processing*, 36(2) :287–314, 1994.

[48] P. Comon, J.-L. Voz, and M. Verleysen. Estimation of performance bounds in supervised classification. In *European Symposium on Artificial Neural Networks, Brussels (Belgium)*, pages 37–42, 1994.

[49] J.-P. Coquerez and S. Philipp. *Analyse d'images : filtrage et segmentation.* Masson, 1995.

[50] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling.* Chapman and Hall, London, 1994.

[51] J. Cutrona, N. Bonnet, and M. Herbin. A new fuzzy clustering technique based on pdf estimation. In *Information Processing and Management of Uncertainty*, pages 225–232, Annecy, France, jul 2002.

[52] D. Defays. An efficient algorithm for a complete link method. *Computer Journal*, 20 :346–366, 1977.

[53] P. Demartines. *Analyse de données par réseaux de neurones auto-organisés.* PhD thesis, Institut National Polytechnique de Grenoble, France, Thèse de doctorat, 1994.

[54] P. Demartines and J. Hérault. Curvilinear component analysis : a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8 :148–154, 1997.

[55] D. DeRidder and R. Duin. Sammon's mapping using neural networks : A comparison. *Pattern Recognition Letters*, 18 :1307–1316, 1997.

[56] P.J. Devijver and J. Kittler. *Pattern Recognition : A statistical Approach.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[57] E. Diday. La méthode des nuées dynamiques. *Revue Statist. Appl.*, 19(2) :19–34, 1971.

[58] E. Diday, J. Lemaire, J. Pouget, and F. Testu. *Eléments d'analyse de données.* Dunod, Paris, 1982.

[59] D. L. Donoho. High-dimensional data analysis : The curses and blessings of dimensionality. In *Aide-mémoire*, August 2000.

[60] J. J. Droesbeke and J. Fine. *Inférence Non Paramétrique.* Ellipse, Paris, 1996.

[61] D. Dubois and H. Prade. Criteria aggregation and ranking of alternatives in the framework of fuzzy set theory. *Fuzzy Sets and Decision Analysis*, 20 :209–240, 1984.

[62] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information Sciences*, 36 :85–121, 1985.

[63] D. Dubois and H. Prade. On the use of agregation operations in information fusion process. *Fuzzy Sets and Systems*, 142 :143–161, 2004.

[64] R.O. Duda and P.R. Hart. *Pattern Classification And Scene Analysis.* Wiley, New York, 1973.

[65] J. P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Review of Modern Physics*, 57 :617–659, 1985.

[66] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, London, 1993.

[67] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[68] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7 (Part II) :179–188, 1936.

[69] D. Fotheringhame and R. J. Baddeley. Nonlinear principal component analysis of neuronal spike train data. *Biological Cybernetics*, 77 :282–288, 1997.

[70] F. Frisone, F. Firenze, P. Morasso, and L. Ricciardiello. Application of topological representing networks to the estimation of the intrinsic dimensionality of data. In *Proceedings of International Conference on Artificial Neural Networks, Paris, France*, 1995.

[71] K. Fukunaga. Intrinsic dimensionality extraction. *Classification, Pattern Recognition and Reduction of Dimensionality in Handbook Of Statistics*, 2 :347–360, 1982.

[72] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Springer, Berlin, 1995.

[73] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Trans. on Information Theory*, 21 :32–40, 1975.

[74] K. Fukunaga and D. R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 20(2) :176–183, 1971.

[75] J. M. Gonzales-Barrios and A. J. Quiroz. A clustering procedure based on the comparison between the k nearest neighbors graph and the minimal spanning tree. *Statistics and Probability Letters*, 62 :23–24, 2003.

[76] J. C. Gower and G. Ross. Minimum spanning trees and single linkage cluster analysis. *Appl. Statistics*, 18 :54–64, 1969.

[77] M. Grabisch. Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems*, 69 :279–298, 1995.

[78] M. Grabisch, S.A. Orlovski, and R.R. Yager. Fuzzy aggregation of numerical preferences. *The Handbook of Fuzzy Sets Series, Vol. 4 : Fuzzy Sets in Decision Analysis, Operations Research and Statistics, R. Slowinski (ed), Kluwer Academic*, pages 31–68, 1998.

[79] M. Grabisch and P. Perny. Agrégation multicritère. *Logique floue, principes, aide à la décision ; B. Bouchon-Meunier, C. Marsala (eds)*, pages 81–120, 2003.

[80] P. Grassberger. An optimized box-assisted algorithm for fractal dimension. *Physics Letters*, A148 :63–68, 1990.

[81] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica*, D56 :189–208, 1983.

[82] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, pages 4–29, 1984.

[83] G. Grinstein, M. Trutschl, and U. Cvek. High-dimensional visualizations. In *Proceedings of the Visual Data Mining workshop, KDD'2001*, San Francisco, California, 2001.

[84] S. Guha, R. Rastogi, and K. Shim. Cure : An efficient clustering algorithm for large databases. *Inf. Syst.*, 26(1) :35–58, 2001.

[85] S. Hader and F. A. Hamprecht. Efficient density clustering using spanning trees. *Unknown*, 0 :0, 2000.

[86] P. Hall and K. Li. On almost linearity of low dimensionality projections from high dimensional data. *Ann. Stat.*, 21(2) :867–889, 1993.

[87] John A. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.

[88] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76 :175–181, 2000.

[89] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406) :502–516, 1989.

[90] C. G. Healey and J. T. Enns. Large datasets at a glance : Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2) :145–167, 1999.

[91] C. G. Healey and J.T. Enns. A perceptual colour segmentation algorithm. Technical Report TR-96-09, University Of British Columbia, 1996.

[92] C.G. Healey. Choosing effective colours for data visualiation. In *Visualization '96, San Francisco, California*, 1996.

[93] C.G. Healey, K.S. Booth, and J.T. Enns. Harnessing preattentive processes for multivariate data visualization. In *Proceedings Graphics Interface '93, Toronto, Canada*, 1993.

[94] M. Herbin, N. Bonnet, and P. Vautrot. A clustering method based on the estimation of the probability density function and on the skeleton by influence zones, application to image processing. *Pattern Recognition Letters*, 17 :1141–1150, 1996.

[95] M. Herbin, N. Bonnet, and P. Vautrot. Multivariate image analysis and segmentation in microanalysis. *Scanning microsc.*, 11 :22, 1997.

[96] M. Herbin, N. Bonnet, and P. Vautrot. Estimation of the number of clusters and influence zones. *Pattern Recognition Letters*, 22 :1557–1568, 2001.

[97] G. T. Herman and H. Lewkowitz. Color scales for image data. *Computer Graphics and Applications*, page 1992, 72–80.

[98] D. Hilbert. Über stetige abbildung einer linie auf ein flächenstück. *Math. Annalen*, 38 :459–460, 1891.

[99] Alexander Hinneburg and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Knowledge Discovery and Data Mining*, pages 58–65, 1998.

[100] F. Höppner and F. Klawonn. Obtaining interpretable fuzzy models from fuzzy clustering and fuzzy regression. In *Proc. of the 4th Int. Conf. on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES) Brighton, UK*, pages 162–165, 2000.

[101] D.F. Hughes. On the mean accuracy of statistical pattern recognition. *IEEE Transaction on Information Theory*, IT 14(1) :55–63, 1968.

[102] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3) :626–634, 1999.

[103] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2 :94–128, 1999.

[104] A. Hyvärinen, J. Karhunen, and E. Oja. *Independant Component Analysis*. John Wiley and Sons, 2001.

[105] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 7(9) :1483–1492, 1997.

[106] A. Hyvärinen and E. Oja. Independent component analysis : Algorithms and applications. *Neural Networks*, 4-5(13) :411–430, 2000.

[107] A. Inselberg. The plane with parallel coordinates. *Special Issue on Computational Geometry : The Visual Computer*, 1 :69–91, 1985.

[108] A. Inselberg. Visualization of high-dimensional data. *Chemometrics and Intelligent Laboratory Systems*, 60 :147–159, 1994.

[109] A. Inselberg. Visualization and data mining of high-dimensional data. *Chemometrics and Intelligent Laboratory Systems*, 60 :147–159, 2002.

[110] A. Inselberg and B. Dimsdale. Parallel coordinates for visualizing multidimensional geometry. *IEEE Transactions Computers*, 18 :401–409, 1987.

[111] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[112] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Computing Surveys*, 31(3) :264–323, 1999.

[113] L. Jimenez and D. Landgrebe. Supervised classification in high dimensional space : Geometrical, statistical and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(1) :39–34, February 1998.

[114] L. Jimenez and D. Landgrebe. Hyperspectral data analysis and supervised feature reduction via projection pursuit. *IEEE Transaction on Geoscience and Remote Sensing*, 37(6) :2653–2667, 1999.

[115] I. T. Jolliffe. *Principal Component Analysis.* Springer Verlag, 1986.

[116] D. Kaplan and L. Glass. *Understanding Nonlinear Dynamics.* Springer-Verlag, 1995.

[117] J. Karhunen. Neural approaches to independent component analysis and source separation. In *Proc. of the 4th European Symposium on Artificial Neural Networks (ESANN'96), Bruges, Belgium*, pages 249–266, 1996.

[118] L. Kaufman and P. Rousseeuw. *Finding Groups in Data : An Introduction to Cluster Analysis.* John Wiley and Sons, New York, 1990.

[119] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives : Preferences and Value tradeoffs.* Wiley, New York, 1976.

[120] D. A. Keim. Pixel-oriented visualization techniques for exploring very large databases. *Journal of Computational and Graphical Statistics*, 5(1) :58–77, 1996.

[121] D. A. Keim. Designing pixel-oriented visualization techniques : Theory and applications. *IEEE Transaction on Visualization and Computer Graphics (TVCG)*, 6(1) :59–78, 2000.

[122] D. A. Keim and H.-P. Kriegel. Visdb : Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14(5) :40–49, 1994.

[123] D. A. Keim and H.-P. Kriegel. Visualization techniques for mining larges databases : A comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), 1996.

[124] B. Kégl. Intrinsic dimension estimation using packing numbers. *Advances in Neural Information Processing, MIT Press*, 15, 2003.

[125] M. Kirby. *Geometric Data Analysis : An Empirical Approach to Dimensionality Reduction and the Study of Patterns.* John Wiley and Sons, N.Y., 2001.

[126] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78 :1464–1480, 1990.

[127] T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, 1995.

[128] Flip Korn, Alexandros Labrinidis, Yannis Kotidis, Christos Faloutsos, Alex Kaplunovich, and Dejan Perkovic. Quantifiable data mining using principal component analysis. Technical Report CS-TR-3754, University of Maryland, College Park, MD, 1997.

[129] F. Kowalewski. A gradient procedure for determining clusters of relatively high point density. *Pattern Recognition*, 28 :1973–1984, 1995.

[130] M.A. Kramer. Nonlinear principal component analysis using auto-associative neural networks. *AICHE J.*, 37 :233–243, 1991.

[131] J.B. Kruskal. Non metric multidimensional scaling : A numerical method. *Psychometrika*, 29 :115–129, 1964.

[132] D. Landgrebe. On information extraction principles for hyperspectral data. In *4th International Conference on GeoComputation*, Fredericksburg, Virginia, USA, 25-28 July 1999.

[133] Larousse. *Le Petit Larousse Illustré 2004.* Larousse, 2003.

[134] P. Lascaux and R. Théodor. *Analyse numérique matricielle appliquée à l'art de l'ingénieur, tome 2 : Méthodes itératives.* Dunod, 2000.

[135] L. Lebart, A. Morineau, and M. Piron. *Statistique exploratoire multidimensionnelle.* Dunod, 2002.

[136] A. Lendasse, E. de Bost, and M. Verleysen. Estimation de la dimension intrinsèque d'une série temporelle et prédiction par une méthode de projection. In *Proceedings of ACSEG'98 - Connectionnist Approaches in economics and Management Sciences, Louvain-la-Neuve (Belgium)*, pages D–37–D–46, 1998.

[137] M. Lennon. *Méthodes d'analyse d'images hyperspectrales. Exploitation du capteur aéroporté CASI pour des applications de cartographie agro-environnemantale en Bretagne.* PhD thesis, Université de Renne I, décembre 2002.

[138] R. D. Luce and H. Raiffa. *Games and Decisions.* Wiley, New York, 1957.

[139] D. J. C. MacKay and M. N. Gibbs. Density networks. In *Statistics and Neural Networks*, pages 129–146. O.U.P., 1998.

[140] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proc. Symp. Math. Statist. and Probability (5th)*, 1 :281–297, 1967.

[141] E. C. Malthouse. Limitations of nonlinear pca as performed with generic neural networks. *IEEE Transaction on Neural Networks*, 9(1) :165–173, 1998.

[142] B. Mandelbrot. *Fractals : Form, Chance and Dimension.* Freeman, 1977.

[143] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 3 :507–522, 1994.

[144] D. De Mers and G. Cottrell. Non–linear dimensionality reduction. In *Advances in Neural Information Processing Systems*, volume 5. Morgan Kaufmann, 1993.

[145] M.C. Minnotte and R. W. West. The data image : a tool for exploring high dimensional data sets. In *Proceedings of the ASA Section on Statistical Graphics*, Dallas, Texas, August 1998.

[146] M. Miyahara and Y. Yoshida. Mathematical transform of (r,g,b) colour data to munsell (h,v,c) colour data. In *Visual Communications and Image Processing '88, SPIE*, pages 650–657, 1988.

[147] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering*, 13(1) :124–141, 2001.

[148] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms which use cluster centres. *Computer Journal*, 26 :354–359, 1984.

[149] G. Nason. Three-dimensional projection pursuit. *Applied Statistics*, 44(4) :411–430, 1995.

[150] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998.

[151] Y. Ohta, T. Kanade, and T. Sakai. Colour information for region segmentation. *Computer Graphics and Image Processing*, 13 :222–241, 1980.

[152] E. Ott. *Chaos in Dynamical Systems.* Cambridge University Press, 1993.

[153] Y. H. Pao and Z. Meng. Visualization and the understanding of multidimensional data. *Engineering Applications of Artificial Intelligence*, 11 :659, 667, 1998.

[154] M. Partridge and R.A. Calvo. Fast dimensionality reduction and simple pca. *Intelligent Data Analysis*, 2 :203–214, 1997.

[155] E. Parzen. On estimation of a probability density function and mode. *Ann.Math.Statist.*, 33 :1065–1076, 1962.

[156] G. Peano. Sur une courbe qui remplit toute une aire pleine. *Math. Annalen*, 36 :157–160, 1890.

[157] K. Pettis, T. Bailey, T. Jain, and R. Dubes. An intrinsic dimensionality estimator from nearest-neighbor information. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25 :165–171, 1976.

[158] R. Pickett and G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics,Beijing and Shenyang, China*, 1988.

[159] C.R. Rao. The use and interpretation of principal component analysis in applied research. *Sankya serie A*, 26 :329–357, 1964.

[160] G. Rellier, X. Descombes, F. Falzon, and J. Zerubia. La poursuite de projection pour la classification d'image hyperspectrale texturée. Technical report, INRIA, mars 2001.

[161] W. Ribarsky, E. Ayers, J. Eble, and S. Mukherjea. Glyphmaker : creating customized visualization of complex data. *IEEE Computer*, 27 :57–64, 1994.

[162] J. B. T. M. Roerdink and A. Meijster. The watershed transform : Definition, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41 :187–228, 2000.

[163] A. Sachinopoulou. Multidimensional visualization. Technical report, Technical Research Centre of Finland, VTT Tiedoteita, Meddelanden, Research Notes 2114, 2001.

[164] H. Sahbi and N. Boujemaa. Validity of fuzzy clustering using entropy regularization. In *IEEE International Conference on Fuzzy Systems (Fuzz'IEEE 2005), Reno, USA, May 22-25*, 2005.

[165] J.W. Sammon. A non linear mapping for data analysis. *IEEE Trans. Comput.*, 18 :401–409, 1969.

[166] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases : The algorithm gdbscan and its applications. *Data Min. Knowl. Discov.*, 2(2) :169–194, 1998.

[167] G. Saporta. *Probabilités, Analyse des données et Statistique*. Technip, 1990.

[168] A. Sasov. Non-raster isotropic scanning for analytical instruments. *Journal of Microscopy*, 165 :289–300, 1992.

[169] L. Savage. *The Fundations of Statistics*. Wiley, New York, 1954.

[170] D. W. Scott and J. R. Thompson. Probability density estimation in higher dimensions. In J. E. Gentle, editor, *Computer Science and Statistics : Proceedings of the Fifteenth Symposium on the Interface, Amsterdam*, pages 173–179. North Holland Elsevier Science Publisher, 1983.

[171] A. K. Sen. Social choice theory. *Handbook of Mathematical Economics*, 3 :1073–1080, 1986.

[172] R.N. Shepard. The analysis of proximities : Multidimensional scaling with an unknown function. *Psychometrika*, 27(2) :125–140, 1962.

[173] R. Sibson. Slink : an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1) :30–34, 1973.

[174] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London New-York, 1986.

[175] J. C. Simon and J. Quinqueton. Une technique de classification utilisant un balayage multidimensionnel de peano. *C.R. Académie des Sciences, Paris*, 286 :655, 1978.

[176] E. Skubalska-Rafajlowicz. The closed curve filling multidimensional cube. Technical report, ICT Technical University of Wroclaw, 1994.

[177] E. Skubalska-Rafajlowicz. Applications of the space filling curves with data driven measure preserving property. *Nonlinear Analysis, Theory, Methods & Applications*, 30(3) :1305–1310, 1997.

[178] R. Spence. *Information visualization.* Addison-Wesley, 2001.

[179] R.J. Stevens, A. F. Lehar, and F. H. Preston. Manipulation and presentation of multidimensional image data using the peano scan. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(25) :520–526, 1983.

[180] W. Stuetzle. Unsupervised learning : estimating the cluster tre of a density by analysing the minimal spanning tree of a sample. Technical report, AT&T Labs - Research, 2000.

[181] F. Tackens. On the numerical detemination of the dimension of an attractor. *Lecture Notes in Mathematics*, 1125 :99–106, 1985.

[182] C. R. Tolle, T. R. McJunkin, and D. J. Gorish. Suboptimal minimum cluster volume cover-based method for measuring fractal dimension. *IEEE Transaction on PAttern Analysis and MAchine Intelligence*, 25(1) :32–41, 2003.

[183] M. Tory and T. Moller. Human factors in visualization research. *IEEE Transaction on Visualization and Computer Graphics*, 10(1) :1–13, 2004.

[184] T. N. Tran, R. Wehrens, and L. M. C. Buyden. Clustering multispectral images : a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 77 :3–17, 2004.

[185] L.A. Treinish and T. Goettsche. Correlative visualization techniques for multidimensional data. *IBM Journal of Research and Development*, 35(1/2) :184–204, 1991.

[186] G. V. Trunk. Statistical estimation of the intrinsic dimensionality of a noisy signal collection. *IEEE Transaction on Computers*, 25 :165–171, 1976.

[187] E.R. Tufte. *The visual Display of Quantitative Information.* Graphics Press, Cheshire, Connecticut, 1983.

[188] E.R. Tufte. *Envisionning Information.* Graphics Press, Cheshire, Connecticut, 1990.

[189] J. W. Tukey. *Exploratory Data Analysis.* Addison Wesley, 1977.

[190] M. Verleysen. Learning high-dimensional data. In *LFTNC 2001 : NATO Advanced Research Workshop on Limitations and Future Trends in Neural Computing, Siena (Italy)*, pages 141–162, 2001.

[191] M. Verleysen. Machine learning of high-dimensional data : Local artificial neural networks and the curse of dimensionality. Technical report, Agregation in higher education thesis, Université Catholique de Louvain, 2001.

[192] P. J. Verveer and R. P.W. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1) :81–86, 1995.

[193] E. M. Voorhees. Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Information Processing and Management*, 22 :465–476, 1986.

[194] M.P. Wand and M.C. Jones. *Kernel Smoothing.* Chapman and Hall, New York, 1972.

[195] C. Ware. *Information Visualization, Perception for Design.* Morgan Kaufmann Publishers, San Francisco, USA, 2000.

[196] A. S. Weigend and N. A. Gershenfeld. *Times Series Prediction : Forcasting the Future and Understanding the Past.* Addison-Wesley, 1994.

[197] B. Wekemans, K. Janssens, L. Vincze, A. Aerts, and J. Heertogen. Automated segmentation of μ-xrf image sets. *X-ray Spectrometry*, 26 :333–346, 1997.

[198] M. A. Wong. A hybrid clustering method for identifying high density clusters. *Journal of Am. Statist. Assoc.*, 77 :841–847, 1982.

[199] Z. Wu and R. Leahly. An optimal graph theoretic approach to data clustering : theory and its application to image segmentation. *IEEE Transaction on pattern analysis and machine intelligence*, 15(11) :1101–1113, 1993.

[200] G. Wyszecki and W. S. Stiles. *Color Science : Concepts and Methods, Quantitative Data and Formulas.* John Wiley & Sons, New York, 1982.

[201] Y. Xu, V. Olman, and D. Xu. Clustering gene expression data using a graph-theoretic approach : an application of minimal spanning trees. *Bioinformatics*, 18 :1–10, 2001.

[202] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1) :183–190, 1988.

[203] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical displays : a general framework for visualization and exploration of large multivariate data sets. *Computer and Graphics*, 27 :265–283, 2003.

[204] J. Yang, M. O. Ward, E. A. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In C. D. Hansen G.-P. Bonneau, S. Hahmann, editor, *Proceedings of the symposium on Data visualisation 2003*, pages 019–028. Eurographics Association, 2003.

[205] L. A. Zadeh. *Fuzzy sets and systems.* J. Fox Polytechnic Press, New York, 1965.

[206] T. Zhang, R. Ramakrishnan, and M. Livny. Birch : An efficient data clustering method for very large databases. In *ACM SIGMOD*, pages 103–114, 1996.