N° d'ordre : 2018

THESE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

pour l'obtention du titre de

DOCTEUR

en

Productique : Automatique et Informatique Industrielle

par

François Esson

Un Logiciel de Visualisation et de Classification Interactives de Données Quantitatives Multidimensionnelles

Soutenue le 1er Juillet 1997 devant la commission d'examen :

MM.

P. Vidal	Président	Professeur à l'USTL
J.G. Postaire	Directeur de recherche	Professeur à l'USTL
J.M. Dirand	Rapporteur	Professeur à l'université de Sherbrooke (Québec)
A. Touzani	Rapporteur	Professeur à l'Ecole d'Ingénieurs de Mohamedia, Rabat (Maroc)
J.M. Toulotte	Examinateur	Professeur à l'USTL
D. Hamad	Examinateur	Maître de Conférence à L'USTL

AVANT PROPOS

Le travail présenté dans ce mémoire a été effectué au Centre d'Automatique de Lille, dirigé par Monsieur Le Professeur Pierre Vidal. Je le remercie de l'accueil qu'il m'a réservé au sein de son laboratoire et de l'honneur qu'il me fait en acceptant la présidence du jury de thèse.

J'adresse mes sincères remerciements à Monsieur Jack-Gérard Postaire, Professeur à l'Université des Sciences et Technologies de Lille, pour son excellent encadrement et son soutient tout au long de mes travaux.

Que Messieurs Jean Marie Dirand, Professeur à l'université de Sherbrooke (Québec), et Abderrahmane Touzani, Professeur à l'Ecole d'Ingénieurs de Mohamedia (Maroc) trouvent ici l'expression de ma reconnaissance pour avoir accepté d'être rapporteurs sur cette thèse.

Je remercie également Denis Hamad Maître de Conférence à l'Université des Sciences et Technologies de Lille, pour sa participation au jury de thèse ainsi que pour son aide occasionnelle tout au long de ma recherche.

Que Monsieur Jean-Marc Toulotte Professeur à l'Université des Sciences et Technologies de Lille trouve ici l'expression de toute ma considération pour l'intérêt qu'il a porté à mon sujet de recherche ainsi que pour sa participation au jury de thèse.

J'exprime également mes chaleureux remerciements à Madame Marie Renée Philippe, de l'Université de Poitiers, pour son aide décisive sur l'étude formelle de validité de la représentation.

Je ne saurais terminer cet avant propos sans adresser mes remerciements les plus sincères à tous ceux qui de près ou de loin m'ont aidé par leur compétence et leur soutient moral dans l'élaboration de ce travail, ainsi que dans mes activités d'enseignement.

I. LA CLASSIFICATION AUTOMATIQUE 5

I.1 Introduction 5

I.2 Généralités 6
I.2.1 Les objets traités et leur description 6
I.2.1.1 Matrice des observations / Matrice de proximité 6
I.2.1.2 Les types des paramètres 7
I.2.2 Définitions de la notion de classe 8

I.3 Les méthodes de classification automatique. 9
I.3.1 Classification Exclusive/ Classification Non Exclusive 10
I.3.2 Classification automatique supervisée (Extrinsèque). 10
I.3.2.1 Approche statistique 11
I.3.2.2 Approche métrique 13
I.3.3 Classification automatique non supervisée (Intrinsèque). 16
I.3.3.1 Les méthodes métriques. 16
I.3.3.2 Les méthodes non métriques 19

I.4 Classification automatique ou Opérateur humain? 23

II. Les représentations planes 24

II.1 Introduction 24

II.2 Généralités 25
II.2.1 Descriptif 25
II.2.2 Utilisation des représentations planes 28
II.2.3 Conclusion 29

II.3 Les différentes méthodes de projection bidimensionnelle 29
II.3.1 Les transformations linéaires 30
II.3.1.1 L'Analyse en Composantes Principales ou ACP 31
II.3.1.2 La projection des moindres carrés (« Least Square Mapping ») 32
II.3.1.3 L'Analyse Discriminante ou AD (algorithmes de *declustering*) 34
II.3.1.4 L'algorithme des projections révélatrices (*Projection Pursuit* ou PP) 37
II.3.2 Les transformations non linéaires 39
II.3.2.1 La projection et la réduction dimensionnelle de Sammon 40
II.3.2.2 Les réseaux de neurones 41
II.3.2.3 La représentation par triangulation 48
II.3.2.4 La distance à deux moyennes 50
II.3.2.5 La représentation des k plus proches voisins (k-NN mapping) 50
II.3.2.6 Les projections géométriques 52
II.3.3 Limitations des algorithmes existants 55

III. LA REPRESENTATION ANGULAIRE 56

III.1 Introduction 56

III.2 Principe de la représentation 57
III.2.1 Calcul des coordonnées de la représentation bi-dimensionelles. 58
III.2.2 Définition du référentiel mobile 60

III.3 Avantages de l'algorithme de représentation angulaire 62

III.4 Etude de la représentation angulaire 63
III.4.1 Validité de la représentation 63
III.4.2 Rappels sur la fonction arccos 65
III.4.3 Propriétés particulières de la représentation angulaire 66
III.4.4 Etude de l'hypersurface antécédent d'une droite de décision dans le plan de représentation 68
III.4.5 Exemples 71

III.4.6 Conclusion 74

IV. LES IMPLANTATIONS DE LA REPRESENTATION ANGULAIRE 75

IV.1 Généralités 75

IV.2 Implantation sous Metawindows : le programme INTERACT 77

IV.2.1 Le retour d'information 77
IV.2.1.1 Mode d'affichage des coordonnées des point et vecteur de référence 77
IV.2.1.2 Affichage des valeurs numériques 79
IV.2.1.3 Visualisation des points de la représentation plane 79
IV.2.1.4 Affichage des paramètres d'une observation 80
IV.2.2 Les fonctions de l'application INTERACT 81
IV.2.2.1 Saisie des fichiers d'observations 81
IV.2.2.2 Modification de l'état du référentiel / Affichage des points. 82
IV.2.2.3 Fonctions de classification interactive 82
IV.2.3 Quelques détails de Programmation 84
IV.2.3.1 Calcul des coordonnées cartésiennes du vecteur de référence 85
IV.2.3.2 Organisation des données 88
IV.2.3.3 Extensions 90
IV.2.4 Limitations d'INTERACT 90
IV.2.4.1 Limitations d'affichage 90
IV.2.4.2 Limitations de taille mémoire 90
IV.2.4.3 L'aspect « fermé » de l'application 91

IV.3 Implantation *MS Windows* : Le logiciel MAP 91

IV.3.1 Les améliorations apportées par le développement sous Windows 91
IV.3.2 - Organisation des données MAP vs INTERACT 92
IV.3.2.1 Gestion de la mémoire 93
IV.3.2.2 Organisation des classes de données / opérations de classement 94
IV.3.3 - Le marquage des points et son utilisation 96
IV.3.3.1 La boite de dialogue de gestion des tags 97
IV.3.4 L'interface graphique 99
IV.3.5 Les fonctionnalités de MAP 100
IV.3.5.1 Rappel des éléments innovants par rapport au prototype 100
IV.3.5.2 Affichage et manipulation de l'état du référentiel mobile 100
IV.3.5.3 Les fonctions de la barre de menu 101
IV.3.6 Exemples d'utilisation 107

V. EXPERIMENTATIONS 111

V.1 Sous l'environnement Metawindows 111

V.1.1 Expérimentation sur un fichier de classes générées artificiellement. 111
V.1.2 Données réelles : Analyse d'une image satellite S.P.O.T. 113

V.2 Sous l'environnement Windows 116

V.2.1 Récapitulatif 116
V.2.2 Etude d'un fichier d'observations apidologiques 116
V.2.2.1 Représentation à l'aide de MAP 116
V.2.2.2 Autres représentations 119
V.2.3 Etude de l'ensemble d'observations « Iris Data » 120
V.2.3.1 Représentation et classification à l'aide de MAP 120
V.2.3.2 Autres représentations 124
V.2.3.3 Conclusion 125
V.2.4 Segmentation d'Images Couleurs par Classification Interactive 126
V.2.4.1 Utilisation du logiciel pour l'aide à la segmentation 126
V.2.4.2 Suppression des points de contours 127
V.2.4.3 Suppression des points achromes 128
V.2.4.4 Traitements des points colores 129
V.2.5 Segmentation interactive d'images brutes 129
V.2.6 Conclusion 131

VI. OUVERTURES / CONCLUSION ____ *133*

VI.1 Ouvertures ____ **133**

VI.1.1 Aide à la décision ____ 133

VI.1.1.1 Choix d'un critère adéquat ____ 133

VI.1.1.2 Mise en œuvre du critère choisi ____ 135

VI.1.1.3 Les algorithmes génétiques ____ 135

VI.1.2 Extensions Possibles des fonctionnalités de l'interface ____ 139

VI.2 Conclusion ____ **140**

VII. BIBLIOGRAPHIE ____ *141*

Tableau des notations :

$\Re$	: ensemble des réels
$\Re^n$	: espace réel de dimension **n**
n	: dimension de l'espace des observations
q	: indice d'une observation
p	: nombre d'observations
K	: nombre des classes
C	: classe d'observations (à modifier)
$\vec{X}$	: une observation multidimensionnelle
$\vec{Y}$	: représentation plane de $\vec{X}$
$\vec{r}_1, \vec{r}_2$	: vecteurs de la base du plan de projection
δ	: distance euclidienne
S	: matrice de covariance globale
$S_{x,k}$	: matrice de covariance de la classe d'ordre k
$\vec{m}_{x,k}$	:moyenne de la classe d'ordre k
B	: matrice de covariance inter-classe $B = (\vec{m}_{x,k} - \vec{m}_{x,l})(\vec{m}_{x,k} - \vec{m}_{x,l})^T$
X	: ensemble, ou matrice des observations multidimensionnelles
ω_k	:classe d'observations d'ordre k
p_i	: nombre d'observation de la classe i
$\vec{m}_x^e$	: moyenne de l'ensemble des observations privé de la frange *e*
S_i	: somme pondérée des sorties des neurones précédant le neurone d'ordre i
$W_{k,i}$	: poids de la connexion qui lie le neurone k, de sortie O_k, au neurone i
L_i	: latitude de déplacement sur l'axe i
$\vec{V}_{ref}$	: vecteur de référence du référentiel de représentation angulaire
0_{ref}	: Origine du référentiel de représentation angulaire

CHAPITRE I

LA CLASSIFICATION AUTOMATIQUE

I. LA CLASSIFICATION AUTOMATIQUE

I.1 Introduction

Le fait de classer des objets en fonction de leurs similarités semble être une démarche innée chez l'homme puisqu'elle est à la base du langage. Par le simple fait de nommer les objets qui nous entourent, on les classe dans des groupes différents. La classification est donc l'une des quêtes scientifiques les plus anciennes de l'homme, ainsi que la base de beaucoup de nos connaissances. Les premiers essais rigoureux de classification[1] que nous connaissons proviennent de la civilisation grecque: citons, en exemple, le système élaboré conçu par Aristote pour classer les espèces animales, et qui fut à la base des tentatives de classification faites par les biologistes au XVIIIe siècle.

Ce sont donc les naturalistes du siècle des lumières qui furent à l'origine des premiers développements de la classification telle que nous la connaissons aujourd'hui. Par la suite, le champ d'application s'est peu à peu élargi : citons en exemple la classification des éléments chimiques par Mendeleyev en 1860, la classification des étoiles naines et géantes, qui utilisait les tracés Température-Luminosité de Hertsprung-Russell, et, à l'orée du XX^{e} siècle, les tentatives de classification en psychologie [BUR09].

De nos jours, de la physique des particules à l'archéologie, de la linguistique à la biologie, de l'économie à la médecine, de nombreuses disciplines couvrant des domaines très divers, sont appelées à manipuler des données de plus en plus volumineuses.

Avant l'avènement des premiers calculateurs vers la fin des années 50, les chercheurs, pour inspecter un ensemble de données multidimensionnelles, n'avaient pour tout outil que celui des représentations par couples de paramètres. Les ordinateur ont permis de faire sauter ces limitations et de mettre en oeuvre des solutions proposées depuis longtemps, mais impossibles à apppliquer sans la puissance de calcul de ceux-ci. La classification, qui est l'un des aspects les plus ambitieux de l'analyse de données, est donc née de la rencon-

---

[1] Au sens large

tre d'un besoin, d'un outil, et de diverses démarches issues de la recherche fondamentale [EVE77][MCD83].

Le concept de « classification » étant en lui-même très général, il nous paraît nécessaire, tout d'abord, de préciser le cadre de notre propos. Ce sera l'objet du paragraphe suivant

I.2 Généralités

I.2.1 *Les objets traités et leur description*

Les algorithmes de classification, sont, bien sûr, adaptés aux types de données à analyser. C'est pourquoi il nous semble nécessaire de préciser les différents types de représentations des données ainsi que les échelles correspondantes. Andenberg, dans son livre "Cluster Analysis for Applications" [AND73], met en avant un inventaire des types de données et des échelles utilisables en classification (Cf. Fig.1)

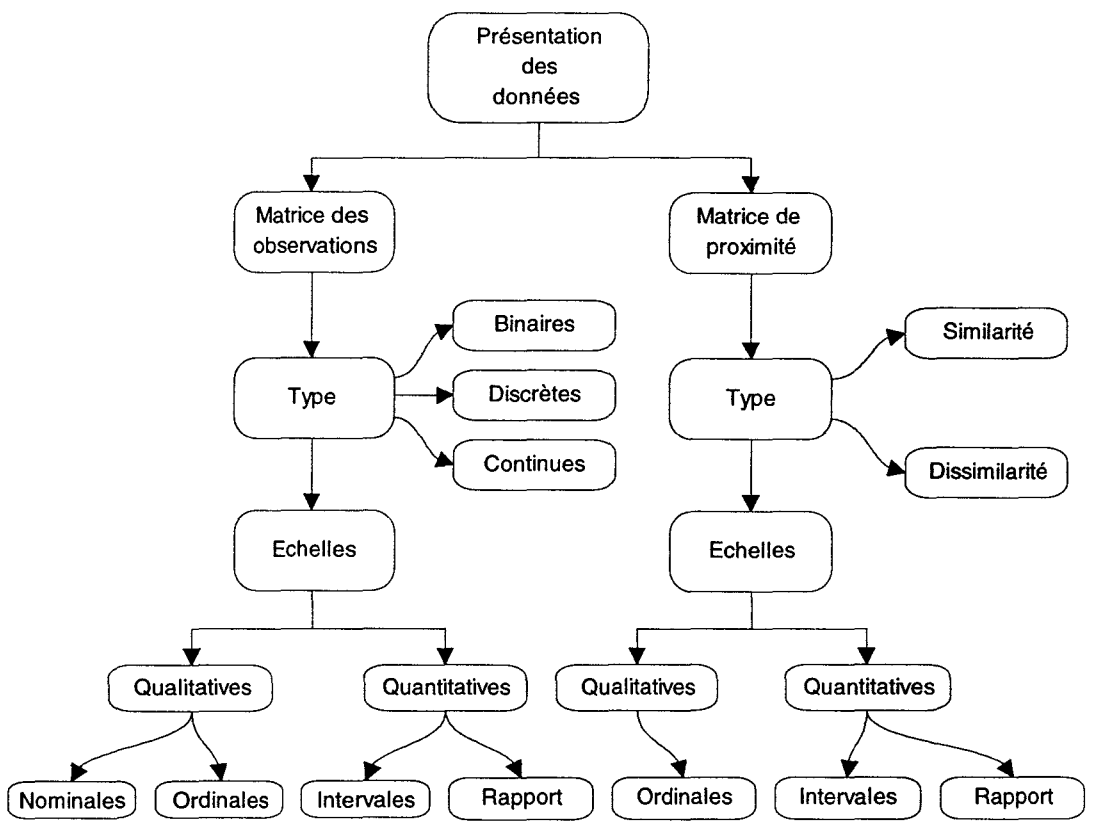


Fig. 1 : Les types de données et leurs échelles

I.2.1.1 Matrice des observations / Matrice de proximité

•Matrice des observation

Soient **p** objets, caractérisés chacun par **n** mesures, paramètres attributs (« features »), où encore composantes[2]. L'ensemble de ces objets peut alors être caractérisé par une matrice **p×n**, que nous appellerons *matrice des observations*[3]. Chaque ligne de cette matrice définit une observation, et chaque colonne, les valeurs que prend un des paramètres mesurés sur l'ensemble des objets.

On représente en général les **n** paramètres comme un ensemble d'axes orthogonaux, de telle sorte que les **p** lignes décrivant les objets peuvent être considérées comme des vecteurs de dimension **n,** et les **p** objets représentés comme des points dans l'espace de dimension **n**.

•Matrice de proximité

Certaines méthodes de classification nécessitent l'établissement d'un indice de proximité, ou de similarité entre des paires d'observations.

Cet indice peut être calculé à partir de la matrice des observations, ou bien élaboré à partir des données brutes. Dans certaines applications psychométriques, les données sont acquises sous formes de proximités. Par exemple, les réponses sont demandée sur une échelle de 0 à 10.

Une matrice de proximité[4] [d(i,j))] regroupe les indices de proximité par paire d'observation. Bien sûr, on ignore la diagonale puisque chaque observation aura le même degré de proximité avec elle-même. Un indice de proximité peut être soit une similarité, soit une dissimilarité.

I.2.*1*.2 Les types des paramètres

Les différents types des paramètres se réfèrent au degré de quantification: un paramètre peut être **binaire**, **discret**, ou bien **continu**. Un paramètre *binaire* ne pourra prendre que deux valeurs et se retrouvera par exemple dans les réponses par oui ou non à un questionnaire. Un paramètre *discret* aura

[2] pour reprendre la terminologie utilisée en imagerie (composante verte, bleue... etc.)
[3] *pattern matrix, dans la terminologie anglo-saxonne*

un nombre fini de valeurs possibles, bien en deçà des capacités numériques maximales du calculateur qui le traitera : par exemple un niveau de gris dans une image numérisée.

Bien sûr, toutes les valeurs stockées dans un ordinateur sont, stricto sensu, discrètes, mais il est souvent plus commode de penser à un paramètre comme à un point sur la droite réelle, ce dernier pouvant prendre n'importe quelle valeur dans un intervalle donné. On qualifiera une tel paramètre de *continu*.

Nous nous placerons, dans le contexte de ce travail, dans le cadre de l'analyse de données multidimensionnelles **quantitatives**, si possible **continues** ou **quantifiées** sur un nombre suffisamment grand de valeurs pour être considérées comme continues. Dans ce cadre, l'objectif de la classification sera de découvrir une organisation sous-jacente aux données étudiées sous formes de classes, et non pas d'établir des lois intrinsèques les régissant, comme on peut le faire dans le cadre des techniques explicatives (régression, décomposition modale, etc.).

I.2.2 *Définitions de la notion de* classe

Une classe est composée d'un certain nombre d'objets similaires que l'on peut grouper ensemble. D'après Brian Everitt [EVE77],

- « Une classe est un ensemble d'entités qui sont *semblables*, alors que des entités provenant de classes différentes ne sont pas semblables »
- « Une classe est un agrégat de points dans l'espace des tests[5], tel que la distance entre deux points de cette classe est moins importante que celle entre n'importe quel point de la classe et n'importe quel point ne lui appartenant pas »
- « Les classes peuvent être décrites comme des régions connexes de l'espace multidimensionnel contenant une relativement grande densité de points, séparées des autres régions de ce type par une région contenant une relativement faible densité de points »

---

[4] proximity matrix dans la terminologie anglo-saxonne
[5] L'espace de test dans la terminologie d'Everitt correspond à l'espace des observations ou des données.

Les deux dernières définitions supposent que les objets à classer sont représentés comme des points dans l'espace multidimensionnel des observations. Nous conserverons cette approche par la suite.

I.3 Les méthodes de classification automatique.

On peut organiser les différents domaines de la classification automatique de multiples manières. Nous le ferons suivant deux critères : supervisé/non supervisé et métrique/statistique, délimitant ainsi quatre sous domaines de recherche. Nous nous baserons sur cette organisation pour poursuivre notre étude. Le diagramme de la figure 2 illustre les différentes catégories que nous aborderons :

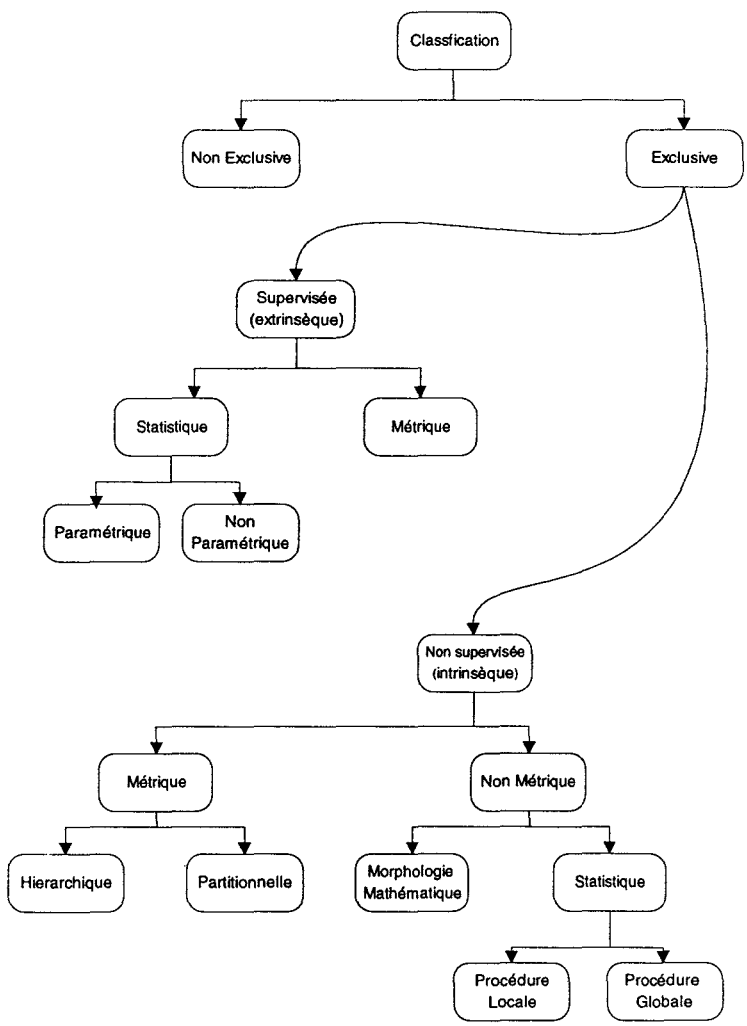


Fig. 2 : Différentes approches du problème de classification

I.3.*1* *Classification Exclusive/ Classification Non Exclusive*

Une classification **exclusive** est une **partition** des objets que l'on étudie. Chaque objet appartient à une classe et une seule, et tous les objets sont classés.

Une classification **non exclusive** peut assigner un objet à plusieurs classes. Par exemple, un **regroupement** d'individus par **âge** ou par **sexe** est **exclusif** alors qu'un regroupement par **catégorie de maladie** est **non exclusif,** puisqu'une personne peut avoir plusieurs maladies simultanément. Shepard et Arabie [SHE79] fournissent plusieurs méthodes de classification non exclusives[6]. La classification floue est un type de classification non exclusive où l'on attribue à un objet, des degré d'appartenance aux classes définies [JAI88].

Le cadre de notre propos est celui où, d'après Everitt [EVE77], un objet appartenant à une classe ne peut appartenir aussi à une autre classe, donc celui de la classification **exclusive**.

I.3.2 *Classification automatique supervisée (Extrinsèque).*

Le but de la classification supervisée est la conception d'un algorithme, ou classifieur, capable d'assigner toute observation inconnue qui lui est présentée à une classe parmi K classes pré-définies.

La classification automatique supervisée comprend donc deux phases :

- une phase d'apprentissage durant laquelle l'espace de représentation des observations est divisé en régions, en utilisant des observations prototypes dont l'appartenance aux différentes classes a été préalablement déterminée par un superviseur. Chaque région correspond à l'une des K classes possibles.
- Une phase de classement pendant laquelle, pour classer un ensemble d'observations inconnues X_q^T, $X_q^T = [x_{1,q}, x_{2,q},\ldots,x_{n,q}]$ **q=1,2,...,p**, le classifieur identifie la région de l'espace dans laquelle chaque observation se trouve et l'assigne ainsi à la classe correspondante.

---

[6] ou avec chevauchement

Pour obtenir de bonnes performances avec cette stratégie de classement, il est nécessaire que l'ensemble d'apprentissage représente fidèlement la population soumise à l'analyse.

L'apprentissage peut se faire selon deux approches :

- La première consiste à utiliser des informations de type ***statistique***, relatives aux différentes classes obtenues sur la base de l'ensemble d'apprentissage. On parle alors de ***classement statistique***.
- La seconde ne fait aucune référence aux notions de probabilité et de distribution des éléments de chaque classe. Les méthodes non statistiques tombant dans ce cadre sont souvent basées sur l'exploitation de la notion de **distance** pour évaluer les similarités entre les observations soumises à l'analyse. Il s'agit alors de méthodes de ***classement métrique***.

I.3.2.1 Approche statistique

L'approche statistique pour aborder le problème du classement automatique part du principe que les observations sont des distributions de points dans l'espace de représentation des données, caractérisées par leurs fonctions de densité de probabilité. Le but recherché est donc de déterminer un certain nombre de caractéristiques statistiques qui permettent de décrire complètement ces fonctions.

I.3.2.1.*a)Approche paramétrique*

L'approche paramétrique consiste à se donner a priori la forme des fonctions de densité de probabilité qui caractérisent les distributions des observations dans l'espace de représentation des données. Nous envisageons donc des procédures de classement faisant appel aux caractéristiques statistiques de la distribution des observations. La théorie de la décision constitue une approche statistique fondamentale des problèmes de classement qui se trouvent posés en termes probabilistes. Elle permet d'effectuer un classement optimal, basé sur la connaissance des probabilités a priori et des probabilités conditionnelles associées à chaque classe. Cependant, en pratique, on ne

dispose pas de ces informations. Elles doivent être estimées à partir de l'ensemble des prototypes de chaque classe.

Si l'on ne possède aucune information sur les observations à classer, on peut supposer qu'elles proviennent d'un mélange de fonctions de densité de probabilité gaussiennes. Les paramètres à estimer sont alors les vecteurs moyennes et les matrices de covariance des distributions associées aux différentes classes, ainsi que les probabilités a priori de ces dernières[DUD73].

Il faut cependant remarquer que le classement sous une hypothèse paramétrique ne sera satisfaisant que dans la mesure où les distributions des observations suivent effectivement des lois gaussiennes [DUD73].

I.3.2.1.b)*Approche non paramétrique*

Dans le cas non paramétrique, aucune hypothèse restrictive n'est faite quant à la loi de répartition des données. Cette approche ne nécessite par conséquent aucune connaissance a priori sur la structure des données à analyser, et donc, la seule information sur laquelle on se base est celle que l'on peut extraire des données elles-mêmes. Les procédures classiques consistent à rechercher les modes de la fonction de densité de probabilité sous-jacente à la distribution des observations [ASS89]. Deux méthodes pour estimer cette fonction viennent généralement à l'esprit :

La fonction de densité de probabilité sous-jacente peut être estimée à l'aide de la méthode de Parzen-Rozenblatt dite **méthode du noyau**, où l'on se fixe un domaine d'estimation [PAR62].

La fonction de densité de probabilité sous-jacente peut aussi être estimée à l'aide de la méthode due à Cover et Hart, dite **méthode des k plus proches voisins** [COV67]. On peut détecter les modes en remontant les pentes de la fonction de densité de probabilité ainsi estimée [KOO76]. Une autre technique consiste à calculer directement le gradient à partir d'un ensemble d'échantillons [FUK75a].

- Dans un contexte totalement différent, sans référence aucune aux notions de probabilité et de distribution statistique des éléments de chaque classe, il existe d'autres méthodes permettant, à partir des obser-

vations de l'ensemble d'apprentissage, de trouver des surfaces de séparation. Parmi celles-ci, une approche très courante est l'approche métrique:

I.3.2.2 Approche métrique

Dans le cadre de cette approche, pour réaliser un classifieur, on peut supposer que les surfaces de séparation sont définies par une équation mathématique dont il s'agit de calculer les coefficients pendant la phase d'apprentissage. Beaucoup de surfaces séparatrices peuvent être envisagées dont les plus simples correspondent au cas linéaire. D'autres surfaces de séparation d'ordre supérieur à 1 peuvent également être utilisées, par exemple des surfaces quadratiques telles que hypersphères, paraboloïdes, ellipsoïdes, etc.. Ces surfaces s'adapteront mieux à la diversité des cas envisagés que celles issues d'un classifieur linéaire, mais le défaut de cette démarche est que le nombre de coefficients à ajuster dépend directement de la dimension de l'espace des observations. C'est la raison pour laquelle on se limite souvent à des surfaces de séparation de type hyperplan. Cette technique est détaillée ci-après:

I.3.2.2.a)*Cas à deux classes*

Considérons un problème à deux classes, $\mathbf{C_1}$ et $\mathbf{C_2}$. On dispose d'un ensemble d'apprentissage de **p** observations $\vec{X}_q$, **q = 1,2,.., p**. Chaque observation appartient à l'une des deux classes. La sortie du classifieur doit prendre la valeur +1 si l'observation qui lui est présentée appartient à la classe 1, ou la valeur 0 si l'observation qui lui est présentée appartient à la classe 2. Les valeurs désirées d_q à la sortie du classifieur sont donc:

$\mathbf{d_q = 1}$ **pour** $\vec{X}_q \in$ **Classe1**

$\mathbf{d_q = 0}$ **pour** $\vec{X}_q \in$ **Classe2**

Une fonction discriminante linéaire est une application de l'ensemble des observations dans l'ensemble des réels. Elle peut s'écrire sous la forme suivante :

$\mathbf{d : \Re^n \rightarrow \Re}$ **avec**

$\mathbf{d}(\vec{X}) = \vec{W}^T \vec{X} + \mathbf{w_{n+1}}$

$\vec{W} = [\mathbf{w_1}, \mathbf{w_2}, .., \mathbf{w_i}, .., \mathbf{w_n}]^T$ est appelé vecteur poids.

On préfère souvent travailler avec des vecteurs d'observation et un vecteur de poids dits étendus :

$$\vec{Y} = \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix}$$

$\vec{W}' = [\mathbf{w_1}, \mathbf{w_2}, .., \mathbf{w_i}, .., \mathbf{w_n}, \mathbf{w_{n+1}}]^T$

La fonction discriminante $d(\vec{X})$ peut alors s'écrire :

$\mathbf{d : \Re^{n+1} \rightarrow \Re}$ **avec**

$\mathbf{d}(\vec{Y}) = \vec{W}'^T \vec{Y}$.

D'un point de vue géométrique, l'apprentissage supervisé consiste à déterminer le vecteur $\vec{W}'$ de telle sorte que l'hyperplan d'équation $d(\vec{Y}) = 0$ sépare les observations des deux classes (Cf. Fig. 3).

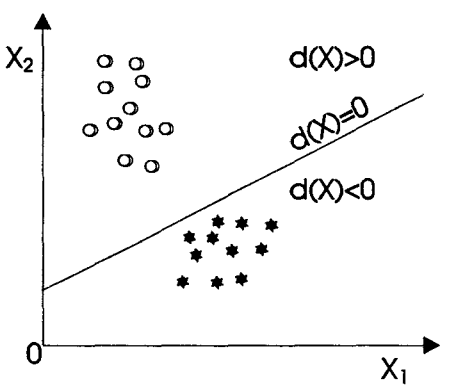


Fig. 3

Si on peut trouver un hyperplan d'équation $d(\vec{X}) = 0$ tel que :

$$\mathbf{d}(\vec{X}) > \mathbf{0} \text{ pour tout } \vec{X} \in \mathbf{Classe1}$$

$$\mathbf{d}(\vec{X}) < \mathbf{0} \text{ pour tout } \vec{X} \in \mathbf{Classe2}$$

$$\mathbf{d}(\vec{X}) = \mathbf{0} \text{ Classe indéfinie,}$$

alors les deux classes sont dites linéairement séparables. La figure 4 montre des classes linéairement séparables (Fig. 4a) et des classes non linéairement séparables (Fig. 4b, Fig. 4c)

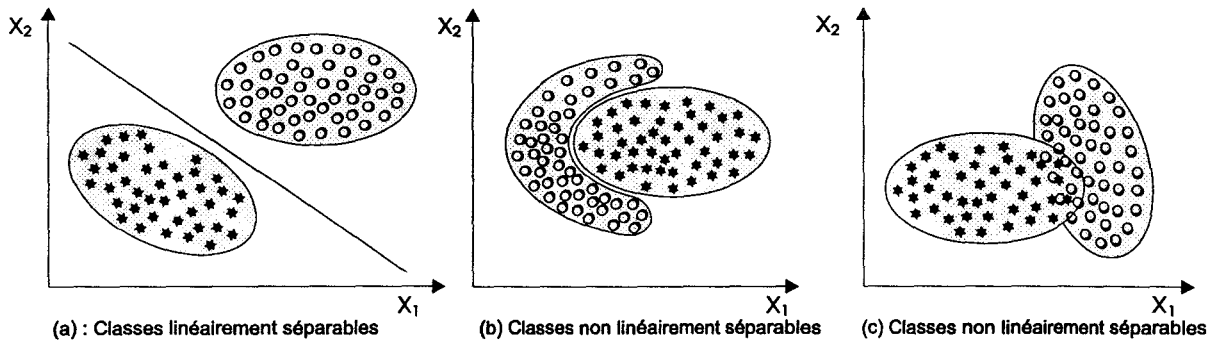


Fig. 4 : différents cas de figure

Il existe plusieurs généralisations directes à plusieurs classes. La plus simple est présentée ci-après, à titre d'information :

I.3.2.2.b)*Généralisation au cas de plusieurs classes*

Un problème à **K** classes , avec **K > 2**, peut être décomposé de manière à être traité comme autant de problèmes à deux classes : On décompose donc le problème global en **K(K-1)/2** sous-problèmes. Nous devons en conséquence définir autant de fonctions discriminantes linéaires d_i, déterminées par $\mathbf{d_k}(\vec{X}) = \vec{W_k}^T \vec{X}$, **k = 1,2,.., K(K-1)/2**

La règle de décision permettant d'assigner une observation $\vec{X}$ à une classe $\mathbf{C_i}$ sera : $\vec{X} \in \mathbf{C_k}$ **si et seulement si** $\mathbf{d_k}(\vec{X}) > \mathbf{d_l}(\vec{X})\ \forall\ \mathbf{l \neq k}$

Plusieurs algorithmes ont été proposés dans la littérature, qui permettent de déterminer des surfaces de séparation linéaires [DUD-73].

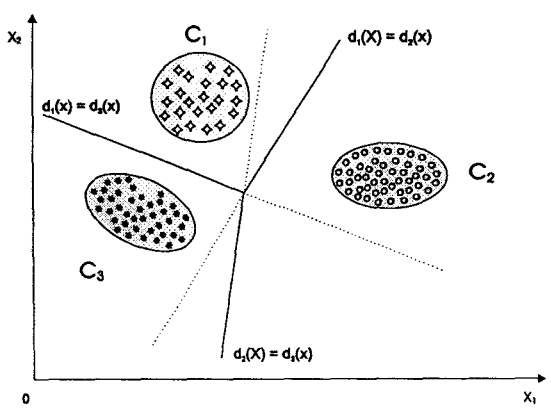


Fig.5 : Surfaces de décision dans le cas de plusieurs classes

I.3.3 *Classification automatique non supervisée (Intrinsèque).*

Alors que la classification supervisée, ou extrinsèque, utilise des objets pour lesquels on connaît les catégories auxquelles ils appartiennent, la classification non supervisée, ou intrinsèque, ne fait appel à aucune connaissance a priori pour détecter des groupements, ou classes, au sein de l'ensemble des données analysées. Dans ce contexte, les procédures se basent soit sur la matrice de proximité, soit sur les observations multidimensionnelles elles-mêmes. En d'autres termes, la classification supervisée se fait en deux étapes: apprentissage/ classement tandis que la classification non supervisée part directement des données pour les classer, sans aucune connaissance a priori.

En effet, il n'est pas toujours possible de disposer d'observations prototypes représentatives de chaque classe. Il est donc impossible, dans de tels cas, de procéder à la phase d'apprentissage qui permettrait de connaître les caractéristiques de ces dernières avant toute tentative de classement.

Il y a bien des façons d'organiser les différents domaines de la classification non supervisée ; nous choisissons de l'aborder en distinguant les approches métriques des procédures non métriques (Cf. Fig. 2).

I.3.3.1 Les méthodes métriques.

I.3.3.1.a)*Généralités*

Une analyse métrique de l'ensemble des données se base sur la notion de distance. Les méthodes entrant dans ce cadre font appel à des notions de similarité entre les individus d'une même classe [SNE73][BAL65]. Il s'agit d'optimiser un critère maximisant la dispersion interclasse tout en minimisant la dispersion intra-classe [FRI67][JON68][FUK70][BAL67][MAC67] [JAI88].

On peut subdiviser le domaine des méthodes métriques de classification non supervisée, en **méthodes de classification hiérarchiques et méthodes de classification partitionnelles**. En fait, la raison pour laquelle on optera

pour l'une ou l'autre de ces approches dépend le plus souvent de la nature des données que l'on analyse.

I.3.3.1.b)*La Classification hiérarchique*

La classification hiérarchique prend comme base de départ, la matrice de proximité. Elle ne fournit pas une classification, c'est à dire une partition définitive de l'espace des observations, mais plutôt une suite de partitions imbriquées les unes dans les autres comme des poupées gigognes. En ceci, une classification hiérarchique est une séquence particulière de classifications partitionnelles.

La classification hiérarchique est particulièrement prisée dans des domaines comme la biologie ou les sciences sociales et comportementales, car elle contribuera par la suite, à la construction d'une taxonomie. Elle est, par contre, peu adaptée aux cas où les données sont en nombre important.

La trace des différentes étapes du processus est un diagramme bidimensionnel structuré comme un arbre inversé, appelé dendogramme.

Les méthodes de classification hiérarchiques sont elles mêmes divisées en deux sous groupes: les méthodes dites agglomératives, et les méthodes dites divisives.

Les méthodes agglomératives procèdent par des séries de fusions successives des **p** données en groupes, le point de départ étant **p** classes contenant chacune une observation, et le résultat final, une seule classe regroupant toutes les observations. Les méthodes divisives ont une démarche duale. Elles procèdent par séries de divisions successives de l'ensemble initial des données constituant la classe de départ, jusqu'à l'obtention de N classes contenant une observation chacune.

Dans tous les cas, le processus de classification fait intervenir l'opérateur, soit dans le choix de paramètres aboutissant à l'arrêt du processus, comme le nombre de classes désirées, soit directement lors de l'examen du dendogramme menant à la classification finale [EVE77][LAW67] [LUK79][BAY80] [SNE73].

I.3.3.1.c)*Les méthodes de classification partitionnelle*

La classification partitionnelle, basée sur l'étude de la matrice des observations, fournit une seule partition définitive de l'espace des observations. Elle sera donc le plus souvent utilisée dans les applications d'ingénierie, où il sera important d'obtenir une partition unique, et sera particulièrement puissante dans le cas où le nombre des observations traitées est important.

Les méthodes de classification partitionnelle sont nombreuses. Nous en citerons deux parmi les plus significatives:

I.3.3.1.c).i *L'algorithme Isodata*

L'algorithme Isodata effectue la classification en assignant tout d'abord les observations à des centres de gravité prédéterminés ou "noyaux", qui évoluent au fur et à mesure que la classification s'affine. Cette classification se fait en cherchant à minimiser un critère de distance entre ces "noyaux" et les observations traitées [BAL67][FRO74].

I.3.3.1.c).ii *La méthode des Nuées Dynamiques*

Cet algorithme a été proposé par Diday [DID71][DID82], et constitue une variante du précédent. Dans celui-ci aussi, les centres de gravité, ou "noyaux" initiaux, peuvent être fixés par l'analyste qui recherche une certaine structure dans l'échantillon qu'il examine. Ils peuvent être aussi issus d'une première phase de classification, basée, par exemple, sur l'exploitation d'une méthode hiérarchique. La méthode des nuées dynamiques consiste alors, tout en minimisant un critère, à recalculer de nouveaux noyaux à chaque itération du processus.

L'algorithme arrive à son terme lorsqu'un critère de stabilité des noyaux est atteint ou lorsque le nombre de classes espéré est obtenu.

I.3.3.2 Les méthodes non métriques

I.3.3.2.a)*Méthodes issues de la morphologie mathématique*

I.3.3.2.a).i *Méthode de morphologie binaire*

La technique de morphologie binaire proposée par C. Botte-Lecocq résulte d'une extension aux données multidimensionnelles d'une approche bien connue en traitement d'image. Les classes d'observations ne sont pas déterminées, comme dans l'approche statistique, à partir des maxima locaux ou des domaines de concavité (§I.3.3.2.b) de la fonction de densité de probabilité sous-jacente de l'ensemble des observations. On adopte, pour leur détection, l'approche de la théorie des ensembles, en transformant l'ensemble fini des observations à analyser en un ensemble discret à valeurs binaires dans l'espace euclidien de représentation des données. Cet espace est discrétisé sur une grille régulière de telle sorte qu'un élément structurant est alors mis en correspondance avec les différents hypercubes de cette grille, où l'on ne prend en compte que la présence ou l'absence d'observations dans chacun des hypercubes. Ce traitement "tout ou rien" est mis à profit pour la classification de l'ensemble des données multidimensionnelles étudié [BOT91][POS93].

I.3.3.2.a).ii *Méthode de morphologie multivaluée*

Dérivée de l'approche précédente, la méthode utilisant les outils de la morphologie multivaluée, proposée par A. Sbihi prend en compte, pour l'extraction des modes de la fonction de densité de probabilité[7] , la densité locale sous-jacente à la distribution des observations estimées en chaque point de l'espace, ce qui affine l'analyse par rapport à une l'approche morphologique binaire[SBI95].

---

[7] en abrégé, la fonction de probabilité sous-jacente à une répartition de points se note fdp (pdf dans la terminologie anglo-saxonne)

I.3.3.2.b) *Méthodes statistiques.*

Les méthodes statistiques constituent un autre courant de la classification automatique multidimensionnelle, souvent associé à l'école Américaine.

Ces méthodes se basent sur la matrice des observations et font en général appel à l'analyse de la fonction de densité de probabilité sous-jacente à la distribution des observations disponibles. On admet alors qu'à chaque mode correspond une classe. La recherche des classes correspond, par conséquent, à une détection des modes de cette fonction [ASS89].

Les méthodes statistiques peuvent être à leur tour schématiquement regroupées en deux catégories de procédures, selon que toutes les observations disponibles sont prises en compte simultanément pour découvrir l'existence de classes ou, qu'au contraire, on s'attache à ne considérer que les relations entre les observations et leurs voisines, pour découvrir la structure de leur distribution.

Ces deux catégories sont connues sous les noms de *Procédures Globales* et de *Procédures Locales*.

I.3.3.2.b).i *Les procédures globales*

Les procédures globales constituent l'approche la plus classique.
On suppose connus les modèles de distribution des observations étudiées. La détermination des classes se ramène ainsi à la détermination des paramètres de chaque fonction de densité de probabilité sous-jacente constituant la fonction de densité de probabilité du mélange.

L'estimation de ces paramètres a été abordée dans la littérature sous deux angles biens connus: les techniques d'apprentissage Bayesien et les procédures d'estimation par maximum de vraisemblance.

- On attribue à R.F. Daly, la formulation Bayesienne de l'apprentissage des paramètres d'un mélange [DAL62]. Cette formulation a été complétée, par la suite, par Hillborn et Lainiotis [HIL68].

- Les premiers à utiliser les procédures d'estimation par maximum de vraisemblance furent Hasselbald [HASS66] puis Day [DAY69], avec des résultats similaires à ceux obtenus par l'approche Bayesienne. D. Cooper et P. Cooper ont aussi proposé des techniques basées sur la détermination des moments des distributions [COO64][COO67].

Le principal écueil de ces méthodes tient au fait qu'il est nécessaire d'établir des hypothèses restrictives pour permettre à ces démarches d'aboutir. Ainsi, le nombre de classes souhaité est souvent un paramètre fixé d'avance par l'utilisateur [SCH76], parfois même restreint à deux [MAK77][MIZ75]. D'autres hypothèses restrictives concernent l'égalité des matrices de covariance, ou la connaissance des probabilité a priori des classes à extraire [WOL70].

I.3.3.2.b).ii *procédures locales*

I.3.3.2.b).ii.1 Extraction des classes par recherche des maxima locaux de la fdp

A l'opposé des méthodes globales, les méthodes locales analysent de manière plus ponctuelle la répartition des observations qui correspondent soit à des classes, soit aux zones de l'espace relativement vides d'observations qui séparent ces classes. La fonction de densité de probabilité sous-jacente peut être évaluée de manière non paramétrique [PAR62][LOF65] afin de mettre en évidence ses maxima locaux.

Il est généralement admis qu'un maximum local d'une fonction de densité de probabilité correspond à une concentration locale d'observations, et donc au centre de gravité d'une classe. L'extraction des classes présentes dans un ensemble de données se réduit donc sous cette hypothèse, à la recherche des modes de la fonction de densité de probabilité.

Plusieurs techniques peuvent être envisagées pour ce faire, à savoir:

- Remonter les pentes de la fdp[7] selon la direction de son gradient [KOO76].
- Déplacer progressivement les observations jusqu'à ce qu'elles atteignent le voisinage d'un mode de la fdp [BOC79].

- Calculer directement le gradient à partir des observations, toujours dans le même but que précédemment [FUK75a].
- Construire une séquence de points par une technique de sériation si bien que la majorité des points voisins d'un mode deviennent des éléments successifs de cette séquence [KIT76].

Mais ces techniques ont un défaut : elles sont très sensibles aux irrégularité locales de la fdp. Elles tendent, de ce fait, à générer de nombreux modes parasites qu'il est par la suite difficile de différencier des modes réels présents.

✍ Une méthode permet de palier ce problème, c'est la technique d'étiquetage probabiliste de Touzani. Celui-ci isole les différents modes de la fdp estimée en effectuant directement une relaxation sur cette dernière [TOU88].

I.3.3.2.b).ii.2 Analyse de la Convexité

Une autre approche consiste à aborder le problème de la détection des modes par le biais de l'analyse de la convexité de la fdp. On peut mener cette approche dans le cadre d'une démarche paramétrique ou non paramétrique. Les modes sont alors assimilés à des régions où la pdf est localement concave. L'analyse de la fdp se fait en l'intégrant sur des domaines d'observation de taille variable [POS82b].

La robustesse est nettement améliorée par opposition aux techniques basées sur l'estimation du gradient, mais cette approche reste sensible à l'ajustement des paramètres de fonctionnement.

Des techniques d'étiquetage probabiliste, ou de relaxation, ont été développées pour améliorer encore la robustesse de cette approche [OLE88].

I.3.3.2.b).ii.3 Extraction des contours de modes

Dans la méthode d'extraction des contours de modes, A. Touzani et J. G. Postaire isolent les modes de la fdp par la détection de leurs contours, effectuée par filtrage médian multidimensionnel de la fdp, et application

d'opérateurs différentiels multidimensionnels [TOU89]. Là aussi, des techniques d'étiquetage probabiliste ont été mises au point afin de réduire la sensibilité de cette approche aux irrégularités locales de la fdp [POS89].

I.4 Classification automatique ou Opérateur humain?

On a vu que toutes les méthodes que nous avons passées en revue font peu appel aux capacités de l'utilisateur. La démarche proposée dans ce rapport consiste, par opposition à intégrer l'opérateur humain dans le processus de classification par le biais d'un dialogue entre l'homme et la machine. Il s'agit de profiter du pouvoir de discrimination visuel de l'opérateur dont la complémentarité avec la puissance de calcul des ordinateurs ne peut qu'augmenter la puissance d'analyse des algorithmes.

Dans le chapitre suivant, nous passons en revue les différentes méthodes de réduction de la dimension et de représentation graphique des données existantes. Ces dernières permettent d'exploiter directement soit dans le processus de classification, soit dans sa validation, le pouvoir de discrimination de l'opérateur humain.

CHAPITRE II

LES REPRESENTATIONS PLANES

II. Les représentations planes

II.1 Introduction

L'examen préliminaire de la plupart des données est facilité par l'utilisation de diagrammes. Les diagrammes ne prouvent rien, mais mettent en lumière les éléments remarquables. Ils ne sont par conséquent pas des substituts pour des tests critiques tels qu'on peut les appliquer aux données, mais ont une grande valeur en suggérant de tels tests, et en expliquant les conclusions fondées sur ces derniers.

R.A. Fisher, *Statistical Methods for Research Workers*, (1925)

Les méthodes graphiques, qui mettent à contribution le discernement de l'opérateur en lui fournissant des supports visuels de l'information sont parmi les plus anciennes dans le domaine des statistiques. On possède des exemples de représentation graphique de données quantitatives datant de trois mille ans avant J.-C. Ces exemples sont les inventaires des troupeaux, récoltes et biens matériels effectués par les scribes de l'Egypte ancienne.

Les potentialités des graphiques sur support papier ont été exploitées de manière exhaustive dans la première partie du XIXe siècle sous la forme d'histogrammes, de représentations planes des nuages de points, etc.. Cependant, cette démarche bien établie, laissait peu de place à des améliorations éventuelles, et jusqu'à une époque récente, le développement de nouvelles méthodes graphiques sortait du cadre principal de la recherche en Analyse de Données. En effet, les nouvelles capacités de calcul rendues possibles par l'avènement des premiers calculateurs ont tout d'abord été exploitées pour la mise en œuvre d'algorithmes de classification automatiques, la représentation devenant par la même accessoire .

Pourtant, la puissance de discrimination du système visuel humain, avec la complexité des processus qu'il met en jeu, reste, encore aujourd'hui, inégalée, et il serait dommage de ne pas chercher à en tirer le plus de profit possible [MCD83][FUM82].

En analyse de données multidimensionnelles, dans la plupart des cas, les objets à classer sont caractérisés par un nombre de paramètres quantitatifs supérieur à trois. Chaque objet est donc représenté par un point dans un espace de dimension **n**, **n** étant le nombre de paramètres mesurés.

L'être humain évolue naturellement dans un espace à trois dimensions ; le concept d'une dimensionnalité supérieure lui échappe. Mais c'est dans un espace à deux dimensions qu'il lui est le plus aisé de raisonner. Par conséquent, si l'on désire fournir à l'opérateur un support visuel de réflexion et d'analyse, il semble logique de tenter de représenter ces données sur un plan, tout en tâchant de conserver la quantité maximum d'information, sachant cependant que toute réduction de la dimension ne peut s'opérer sans une perte d'information.

II.2 Généralités

II.2.*1* *Descriptif*

La transformation d'un ensemble de vecteurs réels de dimension **n** vers un plan, ainsi que le résultat de cette opération, sont appelés *mapping* dans toute la littérature Anglo-saxonne. Il n'existe pas, dans la terminologie francophone, de concept équivalent puisque *mapping* se rapporte à l'idée très générale de cartographie. Nous utiliserons donc les termes approchés de *projection* et de *représentation plane*, usuels dans la littérature francophone ayant trait à l'analyse des données.

L'évaluation de la qualité d'une projection est basée sur le concept de *fidélité*. Celle-ci dépend de la quantité d'information préservée dans le processus de réduction dimensionnelle entre l'ensemble original et sa représentation plane. La *fidélité*, ordinairement évaluée à l'aide de critères mathématiques spécifiques, est une caractéristique essentielle de la plupart des algorithmes de projection. La formulation mathématique d'un critère sera déterminante dans le résultat obtenu et l'usage que l'on pourra en faire en cherchant à préserver telle ou telle propriété de l'ensemble des observations dans leur représentation, ou en orientant la recherche vers un type particulier de répartition (Certains critères, par exemple, favorisent la détection de répartitions « normales » gaussiennes).

D'un point de vue mathématique, ces méthodes de projection sont un cas particulier bidimensionnel de méthodes de réduction dimensionnelle plus générales.

Les méthodes de réduction dimensionnelle permettent de minimiser les problèmes liés au "Curse of dimensionality"[8] que l'on rencontre dans la conception des algorithmes de classification. Ceci est essentiel en analyse de données, où l'on recherche un usage efficace des données en minimisant l'importance des paramètres redondants ou non significatifs. On sait aussi que, pour un même nombre de données, plus la dimension d'un espace est importante, plus celui-ci est *vide*[9].

Cependant, en réduisant la dimension à deux, voire à un, les algorithmes de projection présentent un intérêt supplémentaire par rapport aux méthodes de réduction dimensionnelle, puisqu'ils donnent à l'analyste humain l'accès direct aux données. En cela, les méthodes de projection constituent un ensemble qualitativement distinct de celui des algorithmes qui réduisent la dimension à plus de deux.

Les algorithmes de projection possèdent donc deux atouts majeurs: le fait que ces techniques échappent au problème du « Curse of dimensionality », et la mise à profit des capacités humaines de discrimination.

Le premier groupe de méthodes à voir le jour, était basé sur l'algorithme pour la classification linéaire des données proposé par R.A. Fisher en 1936, et connu sous le nom de "Plan discriminant de Fisher" [FIS36]. Les approches suivantes, qui sont apparues environs trente ans plus tard, furent suscitées par les possibilité graphiques croissantes qu'offraient les gros calculateurs. La première de ces approches est l'algorithme de Sammon, basé sur la préservation des distances inter-points [SAM69]. Cependant, ce dernier rencontre des problèmes de lourdeur de calcul, ce qui a encouragé la poursuite des recherches pour découvrir des algorithmes tout aussi puissants mais plus efficaces.

Aujourd'hui, on peut distinguer pas moins de dix types d'algorithmes de projection plane[10]. La plupart de ces techniques se déclinent en de multiples variantes, si bien qu'il y a en fin de compte plus d'une quinzaine de méthodes recensées à notre connaissance.

---

[8] cette expression fut utilisée pour la première fois par Bellman pour décrire les limitations des méthodes statistiques dont la complexité augmente de manière exponentielle avec la dimension de l'espace des données.
[9] c'est à dire moins la densité globale des données y est importante.
[10] Le notre exclu

Il existe plusieurs façons de classer ces différentes méthodes. L'aspect le plus évident étant, à nos yeux, celui de leur caractère **analytique** ou **non analytique**.

On reprendra par la suite l'approche du paragraphe I-B-1-a, selon laquelle chaque observation multidimensionnelle peut être considérée comme un point dans un espace euclidien, où chaque paramètre constitue un axe,

🕮 Une transformation ***analytique*** est une relation de l'espace multidimensionnel auquel appartiennent les observations (espace de départ), vers l'espace de représentation (espace d'arrivée). Celle-ci transforme donc chaque point de l'espace multidimensionnel de départ en un point dans l'espace d'arrivée (y compris les points n'appartenant pas à l'ensemble fini des observations). La technique n'extraira pas d'information particulière de l'ensemble des points étudié, mais permettra une péréquation directe entre les deux espaces: c'est, en termes mathématiques, une application. Par voie de conséquence, une transformation analytique permet le tracé éventuel dans l'espace multidimensionnel de départ, à partir de frontières de décisions dans le plan de représentation, d'hypersurfaces de décision.

🕮 Une transformation ***non-analytique***, elle, ne portera que sur l'ensemble des observations: c'est une relation d'un ensemble de points multidimensionnels vers un autre ensemble de points, dans notre cas de dimension moindre. La conversion, par définition, ne pourra pas se traduire par une simple expression analytique. Ce type de représentation vise en général à préserver dans la représentation, la structure intrinsèque des observations multidimensionnelles, mais pose souvent un problème qui rend son l'utilisation malaisée dans des applications comme le suivi de processus industriel notamment :

Dans les algorithmes classiques de ce domaine, si une nouvelle observation vient s'ajouter à l'ensemble des données, le calcul de sa représentation fait intervenir la totalité des observations disponibles. Il faut donc recalculer

l'ensemble de la représentation[11]. Ceci se traduit par une certaine lourdeur de calcul, ainsi que par le fait qu'il est impossible d'utiliser ce type de transformation pour l'implantation d'algorithmes de classement. En effet, ces derniers se basent sur un premier calcul des classes existantes à partir d'un échantillon, pour ensuite effectuer le classement à proprement parler des observations (Cf. §I.3.2). Les transformations non-analytiques seront donc généralement utilisées afin d'extraire des données leur structure intrinsèque.

Il s'agit donc de deux groupes de méthodes couvrant des domaines d'intérêts différents, parfois complémentaires.

Notons que la distinction *analytique / non analytique* correspond quasi exactement à la distinction *linéaire / non linéaire*: les transformations non analytiques ne peuvent pas être formulées à l'aide d'équations linéaires par opposition à la plupart des transformations analytiques.

II.2.2 *Utilisation des représentations planes*

Il y a deux applications de base pour les algorithmes de projection : la classification directe et la conception interactive d'algorithmes de classement automatique. Lorsque, par exemple, on définit de manière interactive à l'aide d'une population d'observations types, des frontières de séparation dans l'espace des observations, ou des classes, qui serviront à des algorithmes de classements ultérieurs.

- **La classification directe,** par le biais des représentations planes possède à la fois des avantages et des inconvénients.

  ∞ **Parmi les inconvénients**, citons le fait que chaque type d'algorithme de projection fournit un type de représentation spécifique, ce qui a tendance à influencer le résultat. Quoi qu'il en soit, ceci peut être évité ou atténué, soit par le fait que l'algorithme présente plusieurs représentations distinctes du même ensemble de données, soit par le réexamen des données à l'aide de plusieurs méthodes distinctes, qui à cause de leurs différences structurelles,

---

[11] ce n'est pas le cas des transformations analytiques qui sont des applications de l'espace d'origine dans l'espace de représentation, et donc pour lesquelles le calcul de la représentation d'une observation multidimensionnelle ne fait intervenir que les paramètres de cette observation.

produisent des représentations distinctes. On parlera de ce réexamen sous le terme de *validation croisée*.

ꕥ **Parmi les avantages**, il y a tout d'abord l'aptitude de l'observateur humain à identifier et extraire, à l'aide d'outils graphiques tels que des couleurs différentes pour marquer différentes classes, des groupes indécelables avec des algorithmes de classification classiques. Un autre avantage est que l'inspection visuelle des données donne souvent à l'utilisateur humain, par l'expérience acquise, une meilleure intuition, ainsi qu'une plus grande confiance dans les résultats obtenus par cette technique, plutôt que par une technique de classification classique.

♦ **La conception interactive de classifieurs** est l'autre utilisation des méthodes de mapping. Celle-ci possède à peu près les mêmes avantages et inconvénients que la classification directe. Ajoutons à cela quelques considérations liées à l'estimation du taux d'erreurs des classifieurs sélectionnés. Les seuls estimateurs d'erreur utilisables avec les mappings, sont du type de resubstitution, et (s'il y a un nombre suffisant de données étiquetées), l'estimateur du "hold-out"[BJD81][SIE88]. Ces estimateurs ne sont pas aussi bons que ceux qui sont utilisés pour les méthodes de classification conventionnelles, mais cette situation semble être compensée par la capacité et la tendance du concepteur humain, à créer des hypersurfaces de séparation robustes.

II.2.3 *Conclusion*

A partir de toutes ces considérations, l'utilisation des méthodes de mapping en reconnaissance de formes, et plus spécifiquement en Analyse de Données, semble tout particulièrement prometteuse.

Le paragraphe suivant passe en revue les dix principales techniques de projection, leurs fondements souvent heuristiques, les problèmes d'implantation, ainsi que leurs performances en matière de rapidité de calcul. Nous citerons aussi quels sont les mappings les plus favorables à chaque type de données.

II.3 Les différentes méthodes de projection bidimensionnelle

Les méthodes de projection peuvent se subdiviser en deux grands groupes respectivement de quatre et six sous catégories.

II.3.*1* *Les transformations linéaires*

Les transformations linéaires peuvent se diviser en quatre sous groupes :

1.Les techniques de l'**Analyse en Composantes Principales** (ACP).

2.La projection des **moindres carrés.**

3.Les techniques de l'**Analyse Discriminante** (AD).

4.Les **projections révélatrices** (« *Projection Pursuit* » ou PP).

Les techniques de projection linéaire utilisent des transformations linéaires qui transforment l'espace multidimensionnel $\Re^n$ en un plan.

La projection d'un vecteur $\vec{x}$, de dimension n, vers le plan s 'effectue par une opération matricielle :

$$\vec{y} = A\vec{x} + \vec{b} \qquad \textbf{Equ. 1}$$

où

$$A = \begin{bmatrix} \vec{r}_1^{\,T} \\ \vec{r}_2^{\,T} \end{bmatrix} \qquad \textbf{Equ. 2}$$

Les vecteurs $\vec{r}_1$ et $\vec{r}_2$ sont obtenus en optimisant un critère donné, propre à chaque algorithme de projection. Le vecteur bidimensionnel $\vec{b}$ n'apporte pas ici d'information supplémentaire, mais est donné dans le cadre de la transformation générale. Certaines méthodes lui donnent une valeur nulle, d'autres non nulle.

De par la définition donnée plus haut, toute projection linéaire s'effectuera en deux étapes :

a) le calcul des deux vecteurs de projection $\vec{r}_1$ et $\vec{r}_2$,

b) le calcul de l'image plane de l'ensemble des observations multidimensionnelles.

La seconde partie est commune à toutes les méthodes de projection, alors que la première est spécifique à chaque algorithme.

II.3.*1*.1 L'Analyse en Composantes Principales ou ACP

L'Analyse en Composantes Principales regroupe des méthodes de projections qui utilisent les axes principaux des matrices de covariance définies sur un ensemble de points dans l'espace de dimension **n**. Les directions principales de la matrice de covariance sont définies par les vecteurs propres correspondant à ses plus grandes valeurs propres. Le concept de cette méthode de projection est basé sur la version discrète de l'expansion de Karhunen-Loeve.

Dans le groupe de projections basées sur ce concept, on distingue trois versions, qui diffèrent selon les types de matrices de covariance utilisées pour déterminer les axes principaux.

1. l'ACP *totale* (Selfic) proposée par Watanabe et al. [WAL67],

2. l'ACP classe-conditionnelle (Clafic) présentée en [WAL67],

3. l'ACP classe-conditionnelle standardisée, conçue par Fukunaga et Knootz [FUK70].

On remarque que dans le deuxième et le troisième cas, la représentation nécessite l'existence de prototypes de chaque classe.

- Dans l'ACP totale, les axes de projection sont calculés à partir de la matrice de covariance totale d'un ensemble X de données. Il n'est pas nécessaire d'effectuer un étiquetage préliminaire.
- Dans l'ACP classe-conditionnelle, les axes de projection sont déterminés dans le cadre d'un cas à deux classes, basé sur deux matrices de covariance classe-conditionnelles (non centrées). Un axe de projection est défini par classe d'objets. Par conséquent, deux problèmes indépendants devront être résolus à raison d'un problème par vecteur propre.
- Dans l'ACP classe conditionnelle standardisée, les axes principaux sont déterminés de la même manière que dans le cas précédent. La seule différence est que les vecteurs de projection $\vec{r}_1$ et $\vec{r}_2$ sont calculés à partir du système d'équations suivant (Cf. Equ. 3):

$$\begin{cases} S_{x,1}\vec{r}_1 = \lambda^1 (S_{x,1} + S_{x,2})\vec{r}_1 \\ S_{x,2}\vec{r}_2 = \lambda^2 (S_{x,1} + S_{x,2})\vec{r}_2 \end{cases} \qquad \text{Equ. 3}$$

où $S_{x,1}$ et $S_{x,2}$ sont les matrices de covariance classe-conditionnelles respectives, λ^1 et λ^2 sont les deux valeurs propres les plus grandes, et $\vec{r}_1$ et $\vec{r}_2$, qui décrivent les deux directions principales recherchées, sont les deux vecteurs propres respectifs.

Ces trois versions de la méthode d'ACP sont basées sur la supposition qu'un minimum du carré du critère d'erreur reflète une moindre perte d'information dans le processus de réduction dimensionnelle. Quoi qu'il en soit, ceci ne permet pas d'assurer qu'une projection ainsi obtenue révèle la structure intrinsèque des données et préserve un élément aussi important que la séparabilité des groupes de points. En fait, ce type de projection peut être aisément pris en défaut si les groupes de points ont des formes très éloignées des répartitions gaussiennes, comme des formes de longilignes en dimension 3, par exemple. Dans ce cas, les axes principaux ne préservent pas ces groupes dans la représentation. Ce mode de représentation se montre aussi peu performant si l'ensemble des observations étudié comporte un bruit non négligeable (composantes n'apportant pas d'information supplémentaire).

II.3.1.2 La projection des moindres carrés (« Least Square Mapping »)

Cette technique combine le critère d'erreur quadratique, introduit par Mix et Jones [MIX82], avec une classification hiérarchique agglomerative [SIE88]. C'est donc une technique se basant sur une classification préétablie.

La classification hiérarchique est utilisée pour isoler les groupes de points initiaux dans l'espace des données multidimensionnel, dont la méthode de Mix et Jones, conçue au départ pour l'extraction de paramètres, a besoin.

Un avantage de cette méthode est que l'utilisateur peut tester quasi instantanément une séquence de regroupements offerts par une technique de classification donnée.

Supposons que l'ensemble des données, $X = \{\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_p\}$ contient p observations dans k classes $\omega_1, .., \omega_k$. On spécifie d'abord les centres des classes $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ sur le plan de représentation.

$\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ sont définis comme suit:

$$\vec{u}_j = \begin{bmatrix} \cos(\alpha_j) \\ \sin(\alpha_j) \end{bmatrix}, \text{ avec } \alpha = 2\pi(j-1)/k, \; j = 1, \ldots, k. \qquad \textbf{Equ. 4}$$

Ensuite, connaissant les centres des classes, il reste à calculer la matrice de transformation A, ainsi que le vecteur $\vec{b}$, afin que la représentation des observations, c'est-à-dire l'ensemble des $\vec{Y}_i$ obtenus par la transformation $\vec{Y}_i = A\vec{X}_i + \vec{b}$, soit constituée de nuages de points de variance minimale autour des centres pré-définis.

Cette variance minimale est obtenue si la somme des carrés des distances entre les points du plan appartenant au $j^{ème}$ groupe de points et le centre $\vec{u}_j$ du groupe, est minimisée. Par conséquent, le critère d'erreur quadratique sera le suivant (Equ. 5) :

$$J = \sum_{j=1}^{k} \sum_{\vec{Y}_i : \vec{X}_i \in \omega_j} \left\| \vec{Y}_i - \vec{u}_j \right\|^2 = \sum_{j=1}^{k} \sum_{\vec{Y}_i : \vec{X}_i \in \omega_j} \left\| A\vec{X}_i + \vec{b} - \vec{u}_j \right\|^2 \qquad \textbf{Equ. 5}$$

En dérivant le critère de l'équation 5, par rapport à $\vec{b}$, on obtient

$$\vec{b} = \vec{m}_u - A\vec{m}_x \qquad \textbf{Equ. 6}$$

avec :

$$\vec{m}_u = \left(\sum_{j=1}^{k} p_j \vec{u}_j\right) \Big/ p, \;\; \vec{m}_x = \left(\sum_{i=1}^{p} \vec{X}_i\right) \Big/ p \qquad \textbf{Equ. 7}$$

p_j est le nombre d'observations dans la classe ω_j.

Après avoir substitué l'équation 6 dans l'expression de $\vec{y}_i$, et différencié l'équation 5, cette fois-ci par rapport à A, on obtient la valeur de A, soit

$$A = C_{VW} C_{WW}^{-1} \qquad \textbf{Equ. 8}$$

avec

$$C_{VW} = \sum_{j=1}^{k} \sum_{\vec{x}_i \in \omega_j} (\vec{u}_j - \vec{m}_u) \cdot (\vec{X}_i - \vec{m}_x), \quad C_{WW} = \sum_{j=1}^{k} \sum_{\vec{x}_i \in \omega_j} (\vec{X}_i - \vec{m}_x) \cdot (\vec{X}_i - \vec{m}_x) \qquad \textbf{Equ. 9}$$

Les expressions ci-dessus reprennent les notations utilisées dans [MIX82], mis à part C_{WW} qui est en fait la matrice de covariance du mélange dans l'espace de dimension n, multipliée par le nombre d'observations p.

Le critère d'erreur quadratique employé dans cette méthode est isotropique, et par conséquent, tend à créer sur le plan des répartitions de points de type gaussiennes . Ce plan est le sous espace de l'espace de dimension 2 qui discrimine le mieux les classes de type gaussien. Il risque d'être mis en défaut par des répartitions non gaussiennes, comme des classes de points longilignes par exemple. Cependant, si le nombre de ces classes n'est pas trop important, leur représentation les révélera sous une forme proche de leur forme originale [SIE88].

II.3.*1*.3 L'Analyse Discriminante ou AD (algorithmes de *declustering*)

Ce groupe comprend quatre algorithmes :

1. l'axe discriminant de Fisher (seul $\vec{r}_1$ est utilisé),
2. le plan discriminant optimal,
3. la projection de *declustering*,
4. la projection de *declustering* étendue.

Tous ces algorithmes fournissent une représentation plane des observations dans le cadre d'un problème à deux classes.

• Dans le cas de l'*axe discriminant de Fisher*, Le vecteur $\vec{r}_1$ décrivant l'*axe de Fisher* est obtenu en optimisant le critère suivant :

$$J_F(\vec{r}_1) = \frac{\vec{r}_1^{\,T} B \, \vec{r}_1}{\vec{r}_1^{\,T} S_x \, \vec{r}_1}, \qquad \textbf{Equ. 10}$$

aussi appelé **discriminant de Fisher**, où

$$B = (\vec{m}_{x,1} - \vec{m}_{x,2})(\vec{m}_{x,1} - \vec{m}_{x,2})^T, \quad S_x = S_{x,1} + S_{x,2} \qquad \textbf{Equ. 11}$$

et où $\vec{m}_{x,1}$, $S_{x,1}$, $\vec{m}_{x,2}$, $S_{x,2}$ sont les moyennes estimées et les matrices de covariance des deux classes respectives.

La matrice B est appelée matrice de covariance interclasse. Le vecteur qui maximise J_F est le vecteur propre $\vec{r}_1$, solution de l'équation suivante :

$$B\vec{r}_1 = \lambda S_x \vec{r}_1, \qquad \textbf{Equ. 12}$$

avec

$$\lambda = \frac{\vec{r}_1^T B \vec{r}_1}{\vec{r}_1^T S_x \vec{r}_1} \qquad \textbf{Equ. 13}$$

λ est la plus grande valeur propre de la matrice B.

La solution analytique de ce problème est connue [FIS36] et est donnée par l'équation suivante (Equ. 8) :

$$\vec{r}_1 = S_x^{-1}(\vec{m}_{x,1} - \vec{m}_{x,2}) \qquad \textbf{Equ. 14}$$

$\vec{r}_1$ est le premier vecteur *discriminant*. A $\vec{r}_1$ doit être associé un autre vecteur, afin de produire une représentation bidimensionnelle des données. Ce vecteur peut être fixé de manière arbitraire [FIS36][GEL80].

•La méthode du *plan discriminant optimal* [SAM70b][HUG80] a pour critère de base la même fonction que l'axe de Fisher. Cependant, le second axe de projection sera obtenu à partir de $\vec{r}_2$ qui devra aussi maximiser le discriminant de Fisher avec la condition supplémentaire d'orthogonalité : $\vec{r}_1 \bullet \vec{r}_2 = 0$.

•La méthode de *declustering* a été introduite par J. Fehlauer et B.A. Eisenstein [FEH78]. Cette méthode est basée sur une variante du discriminant de Fisher. En gardant la même symbolique que précédemment, le critère à optimiser est :

$$J_D(\vec{r}) = \frac{\vec{r}^T (B + S_{x,2})\vec{r}}{\vec{r}^T S_{x,1} \vec{r}} \qquad \textbf{Equ. 15}$$

Dans cette représentation, le choix des axes discriminants (décrits par les deux vecteurs propres associés aux deux plus grandes valeurs propres) aura la conséquence suivante sur la représentation bi-dimensionnelle: la représentation de la première classe sera bien ramassée, dense, alors que l'autre sera dispersée dans le plan. Par contre, les deux classes sont bien distinctement représentées.

• La méthode de l'axe discriminant de Fisher, le plan discriminant optimal, et la projection de *declustering* peuvent être considérées comme des cas particuliers d'une méthode plus générale : la projection de *declustering* étendue [GEL80]. Le critère général à optimiser est de la forme :

$$J_{Ex}(\vec{r}) = \frac{\vec{r}^T H_1 \vec{r}}{\vec{r}^T H_2 \vec{r}} \qquad \textbf{Equ. 16}$$

avec

$$H_1 = B + \beta\, S_{x,2}, \qquad H_2 = \begin{cases} S_{x,1} + (1-\beta) S_{x,2}, & \text{pour } 0 \le \beta \le 1; \\ S_{x,1}, & \text{pour } 1 \le \beta \end{cases} \qquad \textbf{Equ. 17}$$

β est un coefficient scalaire de dispersion.

La méthode de *declustering* étendue est capable de produire des projections qui ne sont pas réalisables à l'aide des algorithmes décris précédemment.

Il est possible de créer des projections mixtes, qui seraient composées par exemple de l'axe discriminant de Fisher, et d'un axe du *declustering*, ou bien de deux axes issus du *declustering*, mais avec deux coefficients de dispersion β différents. De plus, on peut imposer la condition que les deux vecteurs de projection soient orthogonaux, c'est-à-dire, que le second vecteur de projection soit calculé de la même manière que dans le cas du plan discriminant optimal.

Il y a aussi moyen de modifier la projection grâce à la possibilité d'inverser l'ordre des classes (ceci étant inutile dans le cas de critères symétriques : J_F et J_D). Cette version encore plus générale est appelée : Projection de *declustering* généralisée.

II.3.1.4 L'algorithme des projections révélatrices (*Projection Pursuit* ou PP)

La méthode des projections révélatrices [FTU74], proposée par Friedman et Tukey, et exploitée sur le système Prim-9 [FTT74] est une méthode basée sur l'heuristique suivante: Une projection révélatrice de la structure interne des données est celle qui produit localement de très petites distances inter-points, tout en maintenant la dispersion globale des données sur le plan.

En d'autre termes, on recherche la projection qui fournit des groupes de points aussi compacts que possible, tout en les séparant le plus possible. Cette heuristique était aussi celle du discriminant de Fisher, cependant, à la différence de ce dernier, cette méthode ne nécessite pas d'étiquetage préliminaire des données.

Afin d'éviter les distorsions éventuelles apportées par ces données marginales, une petite fraction ***e*** des observations qui se trouvent aux deux extrêmes de l'ensemble des points projetés, est exclue du calcul. La présence de (e) en indice fait référence au fait que le calcul a été effectué sur l'ensemble des observation, exclusion faite de cette portion e .

Le critère à maximiser, appelé indice de projection, est le suivant :

$$I(\vec{r}_1,\vec{r}_2) = s(\vec{r}_1)s(\vec{r}_2)d(\vec{r}_1,\vec{r}_2) \qquad \textbf{Equ. 18}$$

On aura donc :

$$s(\vec{r}_i) = \left[\frac{1}{(1-2e)p}\sum_{j=ep}^{(1-e)p}\left[\vec{r}_i\cdot(\vec{X}_j - \vec{m}_x^{(e)})\right]^2\right]^{1/2},\quad \vec{m}_x^{(e)} = \frac{1}{(1-2e)p}\sum_{j=ep}^{(1-e)p}\vec{X}_j \qquad \textbf{Equ. 19}$$

Les $\vec{X}_j, j = 1,\ldots,p$ sont les observations multidimensionnelles, e est la fraction des observation exclue du calcul.

Il est présupposé que les observations ont été préalablement ordonnées en fonction de leur projection $\vec{r}_i\cdot\vec{x}_j$ sur l'axe $\vec{r}_i$ C'est-à-dire :

$$i < j \Rightarrow \vec{r}\cdot\vec{x}_i \le \vec{r}\cdot\vec{x}_j$$

La matrice de covariance définie dans l'équation 19 est une version robuste de la matrice de covariance classique.

Le dernier facteur du critère I est $d(\vec{r}_1,\vec{r}_2)$, qui rend compte de la densité locale des observations projetées. C'est une fonction de proximité moyenne de la forme :

$$d(\vec{r}_1,\vec{r}_2)=\sum_{j>i} f(\rho_{i,j})h(R-\rho_{i,j}) \qquad \textbf{Equ. 20}$$

où h est une fonction échelon unité. La fonction f est choisie monotone décroissante, prenant la valeur zéro pour $\rho=R$. Les fonctions $f(\rho)=R-\rho$ et $f(\rho)=R^2-\rho^2$ sont des fonctions remplissant bien ces conditions.

R est le rayon seuil. Il doit être choisi pour que le nombre moyen d'observations contenues dans la fenêtre de calcul de $d(\vec{r}_1,\vec{r}_2)$, définie par la fonction échelon h, ne soit pas seulement une *petite* fraction de p, mais croisse plus lentement que p, comme log(p), par exemple. Une valeur typique utilisée par Friedman et Tukey est 10 % de l'écart type des données calculé le long du premier (et donc le plus important) axe principal. Le coefficient d'élagage *e* varie de 0.01 à 0.05 suivant la valeur de p, et la régularité souhaitée des données.

Le second vecteur $\vec{r}_2$, à l'instar de l'algorithme du *plan discriminant optimal* (Cf. §II.3.1.3), sera calculé comme le premier vecteur $\vec{r}_1$, avec la condition d'orthogonalité : $\vec{r}_2\cdot\vec{r}_1=0$, pour éviter l'égalité entre les deux vecteurs.

•Remarques

- Cet algorithme, bien que fournissant des projections particulièrement intéressantes, est de fait le plus lent des algorithmes de projection linéaire recensés à ce jour.
- La suppression du facteur d dans l'indice I (Equ. 18) nous ramène à une ACP ordinaire pour $e=0$, et à une forme robuste d'ACP[12] pour $e>0$.

[12] Puisque débarrassée de l'influence des points *marginaux* ou *aberrants*

☙On peut constater que l'indice de l'algorithme des projections révélatrices est un indice heuristique de non normalité [DRO8X].

☙L'algorithme de *Projection Pursuit* a suscité beaucoup d'intérêt et donné suite à de nombreuses recherches le prenant pour base. Citons pour exemples :

- Huber [HUB85], qui généralise l'idée qui sous-tend l'algorithme de *Projection Pursuit*, en suggérant qu'une projection intéressante devra maximiser la non-normalité de la représentation des données.
- Jones et Sibson [JOS87], qui proposent deux indices basés l'un sur l'entropie, l'autre, sur des moments d'ordre 3 et 4.
- Friedman [FRI87], qui reprend l'algorithme en s'attaquant au problème de la trop grande influence du comportement des valeurs extrêmes dans la recherche de la non normalité, en proposant un indice nouveau qui met l'accent sur la partie centrale de l'ensemble des données multidimensionnelles.
- Bandemer et Näther [BAN88], qui approchent l'algorithme de *Projection Pursuit* sous l'angle de la logique floue.
- I.S.Yenyukov [YEN89], qui a étudié un certain nombre de critères spécifiques pour rechercher des structures particulières dans l'ensemble des données analysées, comme les anneaux, les structures de type fractales, etc., à l'aide de l'algorithme de *Projection Pursuit*.

II.3.2 *Les transformations non linéaires*

Par opposition aux transformations linéaires, les transformations *non linéaires* ne fournissent pas une formule simple de transformation par laquelle les données multidimensionnelles seraient projetées sur un plan. En fait, pour de nombreuses projections, il n'y a pas d'expression analytique liant un point de l'espace originel à sa représentation dans le plan.

Nous avons répertorié six types de transformations non linéaires:

1. la méthode de ***Sammon,***

2.les ***réseaux de neurones,***

3.la ***triangulation,***

4.la ***distance à deux moyennes,***

5.la représentation des ***k plus proches voisins,***

6.les **projections géométriques.**

II.3.2.1 La projection et la réduction dimensionnelle[13] de Sammon

La transformation de Sammon [SAM69] est l'un des premiers algorithmes de représentation a avoir été implanté. Cette transformation crée les points $Y_i \ i=1,...,p$ représentés dans le plan, directement à partir des observations multidimensionnelles $X_i \ i=1,...,p$. Il existe donc une correspondance exacte entre une observation multidimensionnelle X_i et sa représentation Y_i. Cependant, les coordonnées des points Y_i dans le plan ne sont pas liées à celles des X_i par une relation analytique. La représentation est basée sur la préservation des distances inter-points. Le critère permettant d'optimiser cette préservation des distances est le suivant (Equ.21):

$$J_S = \frac{1}{\gamma} \sum_{i \neq j} \frac{\left[\delta(X_i X_j) - \delta(Y_i Y_j)\right]^2}{\delta(X_i X_j)}, \qquad \textbf{Equ. 21}$$

avec :

$$\gamma = \sum_{i \neq j} \delta(X_i X_j) \qquad \textbf{Equ. 22}$$

où δ est le carré de la distance Euclidienne. Ce critère, introduit par Sammon, est un cas particulier d'un groupe d'algorithmes de représentation plus généraux, connus sous le nom de *techniques classiques (ou métriques) de réduction dimensionnelle* [DAV83]. Une autre généralisation du critère de Sammon a été apportée par Niemman Weiss [NIW79] et s'écrit comme suit (Equ. 23) :

---

[13] La réduction dimensionnelle ne limite pas le nombre de paramètres de la projection à deux.

$$J_m = \frac{1}{\gamma_m} \sum_{i \neq j} \delta(X_i X_j)^m \left[\delta(X_i X_j) - \delta(Y_i Y_j) \right]^2 , \quad \textbf{Equ. 23}$$

avec $$\gamma_m = \sum_{i \neq j} \delta(X_i X_j)^{m+2} \quad \textbf{Equ. 24}$$

m est un entier signé.

On remarque que le cas particulier $m=-1$ donne le critère de Sammon.

Dans leur étude, Siedlecki, Siedlecka et Slansky [SIE88] trouvent que la méthode de Sammon se heurte à deux écueils[14] :

- ✓ Comme nous l'avons vu en (§ II.2.1), chaque apport d'une nouvelle observation à l'ensemble des données représentées, oblige de recalculer l'ensemble de la représentation, ce qui rend cette méthode particulièrement lourde et difficile à utiliser dans un cas comme celui de la surveillance de processus industriel, par exemple, où les données sont constamment réactualisées.

- ✓ son efficacité décroît lorsque le nombre d'observations à analyser s'accroît, et ceci même dans le cas où il existe des groupements d'observations bien distincts.

- ✓ la représentation est distordue par une dimension trop grande de l'espace de départ, et ceci même si la dimension intrinsèque des données est faible.

II.3.2.2 Les réseaux de neurones

II.3.2.2.a) *Introduction*

Les paragraphes suivants traitent de la recherche menée au Centre d'Automatique de Lille, au sein de l'équipe d'Image et Décision, par MM Mohamed Daoudi, Stéphane Delsert, Mohamed Bétrouni, et Denis Hamad [DAA93][DAB93][DAC93] [HDE96][BDH95] [HAD95] [HBE95][BHD96]. Celle-ci ont pour base les travaux de Rumelhart, Hinton et Williams sur le principe de la rétropropagation appliqué aux réseaux de neurones multicouches [RHW86][LEC86][LIP87][WAS89].

Ces réseaux sont formés d'une couche d'entrée qui reçoit l'information à traiter, en l'occurrence, les données brutes multidimensionnelles, d'une couche de sortie et d'une ou plusieurs couches cachées. Dans une structure classique (*feedforward* ou propagation avant), les connexions sont à sens unique depuis l'entrée vers la sortie à travers les différentes couches (Fig.5).

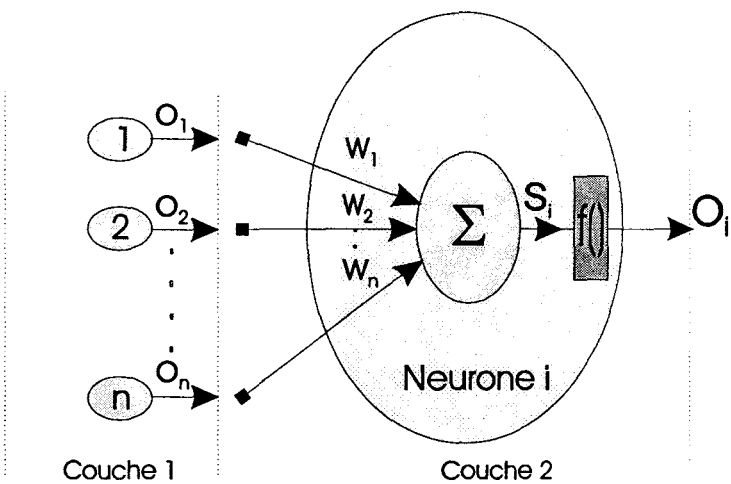


Fig. 5 : Schéma d'une structure neurale classique

Les boucles de retour ainsi que les connections entre éléments d'une même couche ne sont pas autorisées. Chaque sortie O_i d'un neurone i est fonction d'une somme pondérée S_i des sorties des neurones qui le précèdent (Equ.25)(Equ.26). En reprenant la notation de la figure 5 :

$$S_i = \sum_{k=1}^{n} W_{k,i} O_k \qquad \textbf{Equ. 25}$$

$W_{k,i}$ est le poids de la connexion qui lie le neurone k, de sortie O_k, au neurone d'indice i. La sortie du neurone d'indice i est l'application d'une fonction de seuillage sur la sommation, le plus souvent une fonction sigmoïde de S_i, de la forme

$$O_i = f(S_i) = \frac{1}{1+e^{-S_i}} \qquad \textbf{Equ. 26}$$

---

[14] En sus de la limitation mentionnée en II-B-1

Afin de mettre en œuvre ces réseaux pour représenter les observations multi-dimensionnelles, ils ont été organisés de la manière suivante :

La couche d'entrée reçoit les données brutes. Soit

$X = \left[\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_q, \ldots, \vec{X}_p\right]$, la matrice des p observations de dimension n à représenter ; avec $\vec{X}_i = \left[x_{1,i}, x_{2,i}, \ldots, x_{n,i}\right]^T$. La couche d'entrée du réseau est composée de n unités I_k, $k = 1,2,\ldots,n$. Chaque unité I_k est sollicitée par l'attribut $x_{k,q}$ de l'observation $\vec{X}_q$, lorsque celle ci est présentée au réseau.

Dans certains cas, comme l'algorithme qui suit, à cause des contraintes d'homogénéité pendant la phase d'apprentissage, la couche de sortie est également composée de ***n*** unités.

II.3.2.2.b) *Classification Interactive Multidimensionnelle par les Réseaux de Neurones et la Morphologie Mathématique.*

Dans le travail effectué par Mohamed Daoudi [DAA93][DAB93][DAC93], la structure neurale employée comporte une couche cachée, composée de deux unités seulement (Cf. Fig. 6), dont les sorties fournissent la représentation plane.

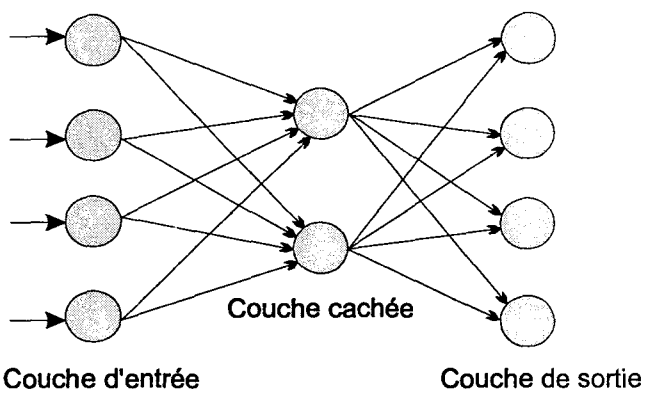


Fig. 6 : Structure du réseau de représentation

C'est la phase d'apprentissage qui détermine la transformation permettant d'obtenir la représentation. Le réseau de neurones est ajusté en utilisant la technique de rétropropagation citée plus haut [RHW86][LEC86] [LIP87][WAS89] en mode auto-associatif. Alors que les **p** observations multidimensionnelles sont présentées successivement au réseau, les poids

des connections sont modifiés de manière itérative, à l'aide de la loi du delta généralisée [RUM86], afin d'obtenir en sortie, une réponse la plus proche possible de l'entrée. Ce processus revient à la minimisation du critère d'erreur suivant, basé sur la distance euclidienne :

$$E = \sum_{q=1}^{p} \left\| \vec{Y}_q - \vec{X}_q \right\|^2 ,$$

où $\vec{Y}_q = \left[y_{1,q}, y_{2,q}, \ldots, y_{n,q} \right]^T$ est le vecteur de sortie du réseau.

Les différentes analyses faites sur le comportement de l'auto-association linéaire et le critère d'erreur [BOU88][BAH89][BAH91] font apparaître le fait que E possède un minimum local et global unique correspondant à la projection orthogonale sur le sous-espace défini par les premiers vecteurs propres de la matrice de covariance associée à l'ensemble d'apprentissage.

Les poids des connections de la couche d'entrée vers la couche cachée et ceux des connections de la couche cachée vers la couche de sortie sont décrits respectivement par les matrices réelles B et A , de dimensions $m \times n$ et $n \times m$, m étant le nombre de neurones de la couche cachée ; en l'occurrence, $m = 2$.

Dans le cas linéaire, l'approche auto-associative donnera :

$A = U \cdot C$ et $B = C^{-1} \cdot U^T$, avec $U = \left[\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m \right]$, matrice formée des m premiers vecteurs propres de la matrice de covariance associée à X avec les valeurs propres ordonnées : $\lambda_1 > \lambda_2 > \ldots > \lambda_m$, et C, une matrice inversible de dimensions $m \times m$. Lorsque C est la matrice unité, les sorties sont données par la relation $\vec{Y} = U^T \cdot \vec{X}$, et cette approche mène aux mêmes résultats que ceux obtenus par l'ACP (Cf. §II.3.1.1).

Cependant, pour la plupart des solutions obtenues en appliquant la méthode du gradient pour minimiser le critère d'erreur, C n'est pas la matrice unité. Ceci est dû au fait que les erreurs répercutées en rétropropagation pendant la phase d'apprentissage sont réparties sur toutes les unités de la couche cachée.

M. Daoudi a associé à cette technique de représentation, deux méthodes de classification automatique à partir de la représentation plane : la méthode Isodata [BAL67], et la méthode de la *ligne de partage des eaux* tirée de la morphologie mathématique [DAB93].

II.3.2.2.c) *Procédures de représentation non-linéaires pour la classification non supervisée.*

Les travaux de Stéphane Delsert et Denis Hamad [HDE96][BDH95] [HAD95] sont basés sur les cartes auto-organisées de Kohonen (Self Organizing Maps ou SOM)[KOH90].

Basés sur la structure neuronale générale décrite plus haut (II-C-2-b-i), les réseaux SOM ont pour but de fournir la représentation bidimensionnelle de l'ensemble des p observations multidimensionnelles données ici sous forme de matrice : $X = \left[\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_q, \ldots, \vec{X}_p\right]$ où $\vec{X}_q = \left[x_{1,q}, x_{2,q}, \ldots, x_{n,q}\right]^T \in \Re^n$ est une observation parmi les p.

Les unités de représentation sont connectées aux unités de la couche d'entrée qui recevront séquentiellement les données brutes à travers des connections pondérées. Le nombre I×J des éléments de la grille de représentation dépend des résolutions horizontale : I et verticale : J .

Chaque unité de représentation est donc caractérisée par sa position (i, j), avec $1 < i < I$, $1 < j < J$, et par son vecteur poids $\vec{W}_{i,j}$, définissant ainsi un champ de réceptivité dans l'espace d'origine des données multidimensionnelles $\Re^n$.

La représentation est générée en établissant une correspondance entre les entrées provenant de l'ensemble des données brutes, et les unités de la grille. La correspondance est ajustée par une séquence d'étapes d'apprentissage. Chaque étape consiste à présenter au réseau un échantillon $\vec{X}(t)$ tiré au hasard parmi l'ensemble X des observations.

On définit le critère $\delta_E(\vec{X}(t), \vec{W}_{i,j}(t))$ comme une mesure de similarité entre $\vec{X}(t)$ et $\vec{W}_{i,j}(t)$. C'est en général une distance euclidienne.

L'unité (i*,j*) dont le vecteur poids $\vec{W}_{i*,j*}$ minimise ce critère est appelée unité gagnante.

Les poids associés à l'unité gagnante ainsi qu'aux unités du voisinage immédiat V_m dans la grille (Equ.28) sont alors mis à jour suivant la relation donnée dans l'équation 27:

$$\Delta\vec{W}_{i,j}(t+1) = \alpha(t)\cdot h_m(i,j)\cdot\left[\vec{X}(t) - W_{i,j}(t)\right] \qquad \textbf{Equ. 27}$$

$$V_m(i^*,j^*) = \left\{(i,j)\,/\,i-m \le i^* \le i+m;\, j-m \le j^* \le j+m\right\} \qquad \textbf{Equ. 28}$$

où m est le facteur de taille du voisinage. V_m couvrira donc $(2m+1)^2$ unités autour de l'unité (i*,j*).

m est fixée à une valeur importante au début de l'ajustement du réseau et décroît jusqu'à zéro au fur et à mesure que celui-ci s'affine. Pratiquement, il décroît linéairement en fonction du nombre d'itérations.

$h_m(i,j)$ est la loi d'interaction. Pour une interaction uniforme, $h_m(i,j)$ est définie par l'équation (Equ. 29)

$$h_m(i,j) = \begin{cases} 1 \text{ si } (i,j) \in V_m(i^*,j^*) \\ 0 \text{ si } (i,j) \notin V_m(i^*,j^*) \end{cases} \qquad \textbf{Equ. 29}$$

$\propto$ est le taux d'apprentissage satisfaisant (Equ. 30).

$$\sum_{t=1}^{T}\alpha(t) \to \infty \text{ et } \sum_{t=1}^{T}\alpha^2(t) \to 0 \text{ ; pour } T \to \infty \qquad \textbf{Equ. 30}$$

Comme les unités d'un voisinage sont mises à jour pendant la phase d'apprentissage, celles-ci ont tendance à représenter les voisinages correspondants dans l'espace multidimensionnel. La topologie des données brutes est donc préservée pendant la projection, et le réseau fournira alors une représentation plane des données.

Pour utiliser cette représentation dans un but de classification interactive, il faut pouvoir fournir à l'utilisateur une représentation graphique des vecteurs poids de chaque noeud de la grille.

Pour cela, on attribue à chaque neurone (i,j) de la grille une valeur G(i,j), valeur médiane des distance entre le vecteur $\vec{W}_{i,j}$ et les vecteurs poids des quatre unités les plus proches [HDE96] (Equ. 31).

$$G(i,j) = Median\left\{\left\|\vec{W}_{i,j} - \vec{W}_{i',j'}\right\|^2 \quad (i',j') \in V_i(i,j)\right\} \qquad \textbf{Equ. 31}$$

Chaque noeud du réseau correspond à un pixel sur l'écran de représentation, auquel est attribué le niveau de gris G(i,j).

Les différents groupes d'observations[15] correspondent alors aux zones claires de l'écran, séparées par des zones plus sombres.

II.3.2.2.d)*L'algorithme de Sammon appliqué aux réseaux de neurones*

Les travaux de MM. M. Bétrouni et D. Hamad [HBE95][BHD96], comme les précédents, s'appuient sur l'architecture neurale décrite en §II.3.2.2.a). Leur algorithme étend l'algorithme de Sammon (§II.3.2.1) aux réseaux de neurones, permettant ainsi une plus grande souplesse dans la représentation, et surtout, évitant l'écueil du calcul systématique de l'ensemble de la représentation à l'apport d'une donnée multidimensionnelle supplémentaire (§II.2.1).

Une première implantation effectuée dans ce but par M.A. Cramer [CRA91] est un réseau à trois couches cachées. La couche cachée centrale comprend m unités utilisées pour la représentation des données, avec $m<n$, n est la dimension de l'espace multidimensionnel des observations.

Le réseau peut être divisé en deux parties principales : la première réalise la projection et la seconde, la *reconstruction*[16]. Le nombre d'unités de la couche cachée est lié à la complexité des relations topologiques entres les données traitées. C'est de ce nombre que dépend la possibilité de généralisation.

Les récents travaux de A. K. Jain et J. Mao [JAM92] proposent une solution alternative basée sur un réseau neuronal à propagation vers l'avant (feedforward), permettant d'éviter ce problème. Ce réseau est composé d'une

---

[15] on parle de « groupes d'observations » car aucune classification n'a été effectuée.

[16] reconstruction de l'ensemble des observations dans leur espace d'origine à partir de leurs projections, afin de pouvoir comparer l'entrée et la sortie du réseau.

couche d'entrée qui reçoit les données brutes, d'une couche cachée, et d'une couche de sortie permettant la représentation des données (Fig. 7).

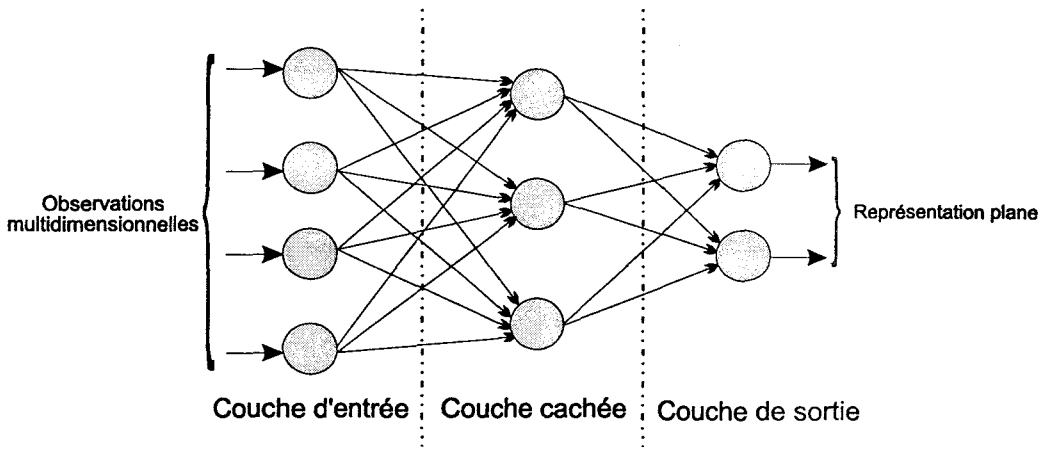


Fig. 7 : Réseau à propagation vers l'avant [JAM92]

La phase d'apprentissage pendant laquelle le réseau s'adapte à la structure intrinsèque des données est de la plus haute importance. La règle d'apprentissage étend l'algorithme de rétropropagation à l'apprentissage non supervisé par le truchement de l'algorithme de Sammon.

C'est dans le cadre de cette dernière approche que se situent les travaux de M. Bétrouni. Il optimise automatiquement le nombre d'unités de la couche cachée à l'aide du critère informationnel d'Akaike [AKA74].

Le nombre d'unités de la couche cachée est modifié et le critère recalculé à l'occasion de chaque phase d'apprentissage. Le nombre d'unités optimal correspond à la valeur minimale du critère. A la fin de l'apprentissage, la partie du réseau effectuant la projection est utilisée pour initialiser le réseau neuronal de A. K. Jain et J. Mao, qui servira à la classification interactive des données par l'utilisateur.

II.3.2.3 La représentation par triangulation

La méthode de représentation par triangulation permet comme, les *projections révélatrices*, (Cf. §II.3.1.4) de fournir à l'utilisateur plusieurs représentations du même ensemble de données.

Soient trois points de l'espace multidimensionnel. Il est toujours possible de les représenter sur un plan de manière à ce que les trois distances inter-points

soient préservées. C'est le principe de base de la représentation par triangulation présentée par Lee, Slage et Blum [LSB79].

Prenons comme point de départ, un ensemble de p observations ordonnées $X = \{\vec{X}_1, \vec{X}_2, ..., \vec{X}_p\}$, avec $\vec{X}_i$ appartenant à un espace métrique de dimension n, muni de la distance δ. Ces observations sont projetées sur le plan de manière à ce que les distances $\delta(\vec{X}_{k-1}\vec{X}_k)$et $\delta(\vec{P}_{ref}\vec{X}_k)$soient exactement préservées. $\vec{P}_{ref}$ est appelé le *point de référence*. A la fin du processus, $2\times p-3$ des $p\times(p-1)/2$ distances seront préservées.

Le point de référence $\vec{P}_{ref}$ pourra être défini de deux manières; ce pourra-t-être :

1.le second plus proche voisin du point représenté, ce qui revient à mettre l'accent sur l'information locale.

2.un point fixe arbitraire donnant lieu à une représentation présentant une faible distorsion globale.

Les auteurs recommandent un ordonnancement des données basé sur le *Minimal Spanning Tree* (MST)[17] construit avec la métrique de l'espace multidimensionnel de départ.

Une fois le MST élaboré, on peut choisir n'importe quel point comme racine, le MST devient alors un arbre orienté, dont les observations sont les sommets. Projeter les points sur le plan revient alors à une procédure de recherche dans une arborescence.

Les auteurs comparent cette démarche à celle d'un laborantin étudiant une préparation sur une lamelle de microscope, et focalisant sur tel ou tel point de l'échantillon observé.

On note que l'on obtient une représentation en forme de spirale si le point d'observation est proche des observations dans l'espace d'origine. Ce phénomène est d'autant plus marqué que les observations sont en nombre important [SIE88].

---

[17] bien connu en classification hiérarchique

II.3.2.4 La distance à deux moyennes

Soient $\vec{m}_{x,1}, \vec{m}_{x,2}$, les deux moyennes estimées des classes ω_1 *et* ω_2, et $S_{x,1}$ *et* $S_{x,2}$, leurs matrices de covariance respectives. Soit $\vec{Y}_i = \left[y_{i,1}, y_{i,2}\right]^T$, une représentation plane de l'observation multidimensionnelle $\vec{X}_i$.

$$y_{ij} = \delta_M\left(\vec{X}_i, \vec{m}_{x,j}, S_{x,j}\right),\ j = 1,2, \qquad \textbf{Equ. 32}$$

δ_M est la distance de Mahalaenobis

$$\delta_M\left(\vec{x}, \vec{m}_x, S_x\right) = (\vec{x} - \vec{m}_x)^T S_x^{-1} (\vec{x} - \vec{m}_x) \qquad \textbf{Equ. 33}$$

Il est à noter que la première bissectrice de la représentation plane sera l'image d'un hyperplan orthogonal au vecteur $\vec{m}_{x,1} - \vec{m}_{x,2}$ si les deux matrices de covariance sont égales, d'une hypersurface quadratique séparant les deux classes dans le cas contraire.

II.3.2.5 La représentation des k plus proches voisins (k-NN mapping)

La représentation des k plus proches voisins [FUM82][FUM84] est basée sur l'estimation des *fdp* $p(x|\omega_i)$ des classes ω_i, $i = 1,2,\ldots,K$, par la méthode des k plus proches voisins (Cf. §I.3.3). C'est donc uniquement un moyen de visualiser une classification déjà effectuée. Lorsqu'on veut classer les observations sur lesquelles ont été estimées ces *fdp*, on effectue la comparaison des probabilités a posteriori, et on obtient l'équation suivante en reprenant la notation vectorielle précédente (Equ.27) :

$$p(\omega_i/\vec{X}) \geq p(\omega_j/\vec{X}) \quad \text{si et seulement si } \delta(\vec{X}, \vec{X}_{(i)}) \leq \delta(\vec{X}, \vec{X}_{(j)}) \qquad \textbf{Equ. 34}$$

$\vec{X}_{(i)}$ est le k plus proche voisin de $\vec{X}$ dans la classe ω_i et $\vec{X}_{(j)}$ est le k plus proche voisin de $\vec{X}$ dans la classe ω_j. Cette règle de classification équivaut à comparer les distances à chaque plus proche voisin à l'intérieur de chaque classe. C'est dans le même ordre d'idée que se situe la représentation des plus proches voisins.

•Pour un problème à deux classes, la représentation plane $\vec{y}_i$ de l'observation multidimensionnelle $\vec{x}_i$ sera définie telle que dans l'équation suivante (Equ. 35) :

$$\vec{Y}_i = \begin{bmatrix} \delta(\vec{X}_i, \vec{X}_{(1)}) \\ \delta(\vec{X}_i, \vec{X}_{(2)}) \end{bmatrix} \qquad \textbf{Equ. 35}$$

Dans ce cas simple, les coordonnées de toute représentation seront simplement les deux distances au plus proche voisin de chacune des deux classes.

•Pour un problème à plus de deux classes et en reprenant la même terminologie que précédemment, la représentation sera définie comme suit (Equ. 36 à 39)

Soit $\vec{X}_i \in \omega_s$, alors

$$\vec{Y}_i = r(\vec{X}_i) \begin{bmatrix} \cos(\alpha(\vec{X}_i)) \\ \sin(\alpha(\vec{X}_i)) \end{bmatrix} \qquad \textbf{Equ. 36}$$

$$r(\vec{X}_i) = \frac{\delta(\vec{X}_i, \vec{X}_{(s)})}{\delta(\vec{X}_i, \vec{X}_{(t)})} \quad \text{avec} \quad 0 < \delta(\vec{X}_i, \vec{X}_{(t)}) \leq \delta(\vec{X}_i, \vec{X}_{(j)}), \ \forall j / j \neq s \qquad \textbf{Equ. 37}$$

On remarque que si $r(\vec{X}_i) > 1$, alors $\vec{X}_i$ est mal classé puisque il devrait être assigné à la classe t.

$$\alpha(\vec{X}_i) = \alpha_0 \frac{\delta(\vec{X}_i, \vec{X}_{(s)})}{\delta(\vec{X}_i, \vec{X}_{(u)})}, \quad 0 < \delta(\vec{X}_i, \vec{X}_{(u)}) \leq \delta(\vec{X}_i, \vec{X}_{(j)}), \forall j / j \neq s,t \qquad \textbf{Equ. 38}$$

$$\alpha_0 \text{ est choisi tel que } 2\pi(s-1) \leq \alpha(\vec{x}) < 2\pi s \ \ \forall \vec{X} \in \omega_s \qquad \textbf{Equ. 39}$$

On trouvera des exemples de représentation par cette méthode dans [SIE88].

II.3.2.6 Les projections géométriques

La méthode des projections géométriques est une méthode proposée par D. Sudhanva et K. Chidananda Gowda [SUG92]. Elle consiste à réduire la dimension de l'espace original de dimension *n* jusqu'à la dimension 2 par projections géométriques successives[18]. La réduction s'effectue comme suit (en gardant la notation précédente) :

Appelons *A* la matrice de l'ensemble des *p* données de dimension *n*. *A* est donnée dans l'équation 40:

$$A = \begin{bmatrix} \vec{X}_1^T \\ \vec{X}_2^T \\ \vdots \\ \vec{X}_p^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pn} \end{bmatrix} \qquad \textbf{Equ. 40}$$

L'observation de rang i sera donc représentée par $\vec{X}_i^T$ et le paramètre d'ordre j de l'observation i sera x_{ij}. Les minima et maxima par paramètre sont (Equ.41) :

$$\begin{aligned} \min j &= \min(x_{j1}, x_{j2}, \ldots, x_{jn}) \\ \max j &= \max(x_{j1}, x_{j2}, \ldots, x_{jn}) \end{aligned} \qquad \textbf{Equ. 41}$$

Les ***p*** observations de dimension ***n*** sont d'abord normalisées selon l'équation 42 pour donner la matrice normalisée A^*.

$$x_{ij}^* = (x_{ij} - \min j)/(\max j - \min j) \quad j = 1 \ldots p, \quad i = 1 \ldots n \qquad \textbf{Equ. 42}$$

et les éléments de la matrice A^* ont dès lors des valeurs de 0 à 1.

Les moyennes et variances par paramètre sont (Equ. 43) :

$$m_j = (1/p)\sum_{i=1}^{p} x_{ij}, \quad s_j = \left[(1/p)\sum_{i=1}^{p} (x_{ij} - m_j)^2 \right]^{1/2}. \qquad \textbf{Equ. 43}$$

Le vecteur des variances par paramètre est donc $\vec{S} = [s_1, s_2, \ldots, s_n]$.

---

[18] Il est bien sûr possible d'arrêter la réduction dimensionnelle à une valeur supérieure à deux.

Les colonnes de la matrice de donnée normalisée A^* sont alors réorganisées par ordre de variance décroissante en fonction de la variance par paramètre précédemment calculée, en prenant pour principe que les paramètres dont la variance est faible constituent une source d'information moins importante que ceux dont la variance est élevée.

$$A^* = \begin{bmatrix} \vec{X}_1^{*T} \\ \vec{X}_2^{*T} \\ \vdots \\ \vec{X}_p^{*T} \end{bmatrix} = \begin{bmatrix} x_{11}^* & x_{12}^* & \cdots & x_{1n}^* \\ x_{21}^* & x_{22}^* & \cdots & x_{2n}^* \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1}^* & x_{p2}^* & \cdots & x_{pn}^* \end{bmatrix}$$

La nouvelle matrice réordonnée est notée B telle que :

$$B = \begin{bmatrix} \vec{Y}_1^T \\ \vec{Y}_2^T \\ \vdots \\ \vec{Y}_p^T \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{p1} & y_{p2} & \cdots & y_{pn} \end{bmatrix} \qquad \textbf{Equ. 44}$$

Le modèle de projection en perspective utilisé pour réduire la dimension de l'espace de départ est donné par l'équation 45:

$$\begin{bmatrix} y_{ij}^{(1)} \\ y_{ij-1}^{(1)} \\ y_{ij-2}^{(1)} \end{bmatrix} = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & n \end{bmatrix} = \begin{bmatrix} y_{ij} \\ y_{ij-1} \\ y_{ij-2} \end{bmatrix} \qquad \textbf{Equ. 45}$$

avec : $(y_{ij-1}^{(1)}, y_{ij}^{(1)})$: projection plane de $(y_{ij-2}, y_{ij-1}, y_{ij})$ et $y_{ij-2}^{(1)} = 0$

L'indice indique l'étape du processus itératif de réduction dimensionnelle.

La réduction dimensionnelle est obtenue de la manière suivante : on projette en perspective (Equ.45) des sous-ensembles de paramètres de dimensions 3 vers le plan, puis les nouveaux paramètres obtenus sont groupés par trois. On réitère le processus jusqu'à obtenir la dimension voulue[19].

Pour réduire progressivement la dimension des données à deux, les auteurs proposent deux méthodes itératives :

- l'écrasement arrière :

[19] deux dimensions dans notre cas

- l'écrasement des triplets.

II.3.2.6.a) *Ecrasement arrière*

Prenons pour exemple un espace des observations de dimension 5.

La i^{ème} observation de l'ensemble réordonné $(y_{i1}, y_{i2}, y_{i3}, \ldots, y_{ij}, \ldots, y_{in})$ sera réduite en utilisant la projection décrite dans l'équation 45, selon le processus itératif donné dans l'équation 46.

En gardant la notation de l'équation 45, on obtient:

$$\begin{aligned} &[y_{i1}y_{i2}(y_{i3}y_{i4}y_{i5})] \Rightarrow [(y_{i1}y_{i2}(y_{i3}^{(1)}y_{i4}^{(1)})] \\ &[y_{i1}(y_{i2}y_{i3}^{(1)}y_{i4}^{(1)})] \Rightarrow [y_{i1}(y_{i2}^{(2)}y_{i3}^{(2)})] \\ &[(y_{i1}y_{i2}^{(2)}y_{i3}^{(2)})] \Rightarrow [y_{i1}^{(3)}y_{i2}^{(3)}] \end{aligned} \qquad \textbf{Equ. 46}$$

A chaque écrasement, une erreur est introduite correspondant à la perte de l'information de profondeur quand on représente un objet en perspective.

II.3.2.6.b) *Ecrasement par triplets*

Dans le cas de l'écrasement par triplets, en reprenant l'exemple précédent, le processus itératif est le suivant (Equ.47) :

$$\begin{aligned} &[(y_{i1}y_{i2}y_{i3})(y_{i4}y_{i5})] \Rightarrow [(y_{i1}^{(1)}y_{i2}^{(1)})(y_{i4}y_{i5}))] \\ &[(y_{i1}^{(1)}y_{i2}^{(1)}y_{i3})y_{i5}] \Rightarrow [(y_{i1}^{(2)}y_{i2}^{(2)})y_{i5}] \\ &[y_{i1}^{(2)}y_{i2}^{(2)}y_{i5}] \Rightarrow [(y_{i1}^{(3)}y_{i2}^{(3)})] \end{aligned} \qquad \textbf{Equ. 47}$$

Les paramètres sont groupés par triplets réduits en doublés, les paramètres restants étant laissés de côté pour être traités à l'opération suivante.

II.3.3 *Limitations des algorithmes existants*

Parmi toutes les différentes méthodes que nous avons vues, peu d'entre elles peuvent réellement servir à la classification d'un ensemble d'observations brutes, sans que l'utilisateur ne possède aucune information a priori sur ce dernier. En effet, la plupart ne servent qu'à visualiser un ensemble d'observations sur lequel une classification a déjà été effectuée :

- ☞ La représentation des plus proches voisins ne sert qu'à mettre en valeur une classification effectuée par la méthode du même nom. Il en va de même pour la représentation des deux moyennes, ainsi que pour les méthodes basées sur l'analyse discriminante.
- ☞L'algorithme de Sammon semble fournir de bons résultats, mais, nous l'avons vu, il présente des lourdeurs d'implantation.
- ☞Les réseaux de neurones permettent de palier ces lourdeurs, mais demandent toutefois un nombre important de calculs ; or ceux-ci sont gourmands en espace mémoire ainsi qu'en ressources système.
- ☞Mises à part les deux méthodes des projections révélatrices et de la représentation par triangulation, la plupart des méthodes existantes ne proposent en tout et pour tout qu'une seule représentation de l'ensemble des données.

Ce sont ces divers écueils que nous nous sommes efforcés de palier. Nous avons donc défini une méthode interactive à la fois rapide, peu gourmande en ressources, et versatile, permettant d'enrichir les possibilités d'inspection des données et de classification de l'utilisateur, sans pour autant requérir de la part de ce dernier de connaissances particulières en Analyse de Données. Cette méthode fait l'objet du chapitre suivant.

CHAPITRE III

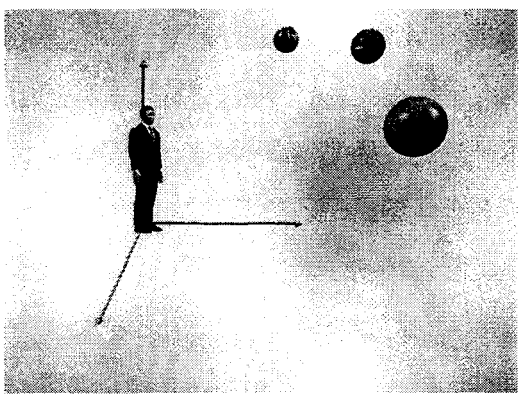
LA REPRESENTATION ANGULAIRE

III. LA REPRESENTATION ANGULAIRE

« Nous avons ce gigantesque pouvoir de mémoriser les images visuelles. Aucune machine n'est capable d'emmagasiner ne serait-ce qu'une fraction des images contenues dans notre tête. »

E. Bizzi, directeur du département d'études sur les sciences cognitives du MIT

III.1 Introduction



Dans bien des domaines scientifiques, et en particulier dans le domaine de l'Automatique, de la Robotique, et de l'Informatique Industrielle, la démarche des chercheurs est souvent caractérisée par une approche anthropomorphique des problèmes.

C'est bien le même type de démarche que l'on retrouve dans des méthodes de représentation de données telles les *projections révélatrices* (II-C-1) ou la méthode de *représentation par triangulation* (II-C-2). En effet, ces deux algorithmes laissent à l'utilisateur le libre choix de la représentation des données qu'il désire étudier. A l'opposé, la plupart des autres méthodes, lorsqu'elles ne sont pas uniquement l'illustration de résultats de méthodes de classification automatique comme la *représentation des plus proches voisins* ou *k-NN mapping* ou la *distance à deux moyennes* (II-C-2), ne fournissent au spécialiste qu'une et une seule représentation, évaluée selon un critère mathématique souvent heuristique comme étant la meilleure.

Le fait de postuler qu'il existe une représentation unique qui soit « meilleure » que toutes les autres est discutable. En effet, ne serait-ce qu'en trois dimensions, pour appréhender un objet en volumes de manière juste, il est nécessaire d'observer ce dernier sous plusieurs angles, et ce, malgré une vision tridimensionnelle binoculaire. Cette démarche est d'autant plus nécessaire dans le cas d'une vision d'où la notion de relief est absente.

Afin de recueillir le maximum d'informations sur les données lors de la visualisation, il serait intéressant que celle-ci soit multiple, voire dynamique et

non unique. C'est en effet par l'acquisition d'un enchaînement d'images successives en constante évolution, que l'homme parvient à modéliser son environnement, à s'en faire une représentation mentale.

Notre technique de représentation découle directement de ces réflexions.

III.2 Principe de la représentation

Considérons la manière avec laquelle l'homme appréhende visuellement son environnement. L'individu peut se déplacer à l'intérieur de l'espace pour modifier son point de vue. Il peut aussi modifier l'orientation de son regard. Ainsi, en acquérant plusieurs images distinctes de son environnement, il effectue une synthèse des informations acquises, et reconstitue, en fin de compte, une image globale cohérente de celui-ci .

L'utilisateur doit donc pouvoir *évoluer* dans l'espace de dimension **n** et choisir un *point d'observation* pour ses données, ainsi que la direction de son *regard.*

La stratégie que nous proposons consiste à choisir, de manière intuitive, le référentiel, que nous appellerons par la suite « référentiel mobile », par rapport auquel la représentation plane est calculée. En conséquence, notons que cette démarche ne nécessite le recours à aucun critère d'optimisation basé sur des approches métriques ou statistiques, comme nous l'avons vu dans le chapitre précédent. Nous faisons appel, dans un premier temps, au discernement et à l'expérience acquise par l'analyste lui-même, qui modifie le référentiel mobile lié à la représentation par essais successifs, jusqu'à ce que cette dernière révèle la structure des données analysées [ESS92].

Il va sans dire que l'outil développé dans le cadre de cette étude doit être extrêmement convivial afin que l'utilisateur puisse en tirer aisément le maximum de profit.

III.2.1 *Calcul des coordonnées de la représentation bi-dimensionelles.*

Imaginons un observateur dans l'espace tridimensionnel. Celui-ci est caractérisé par sa position et la direction de son regard.

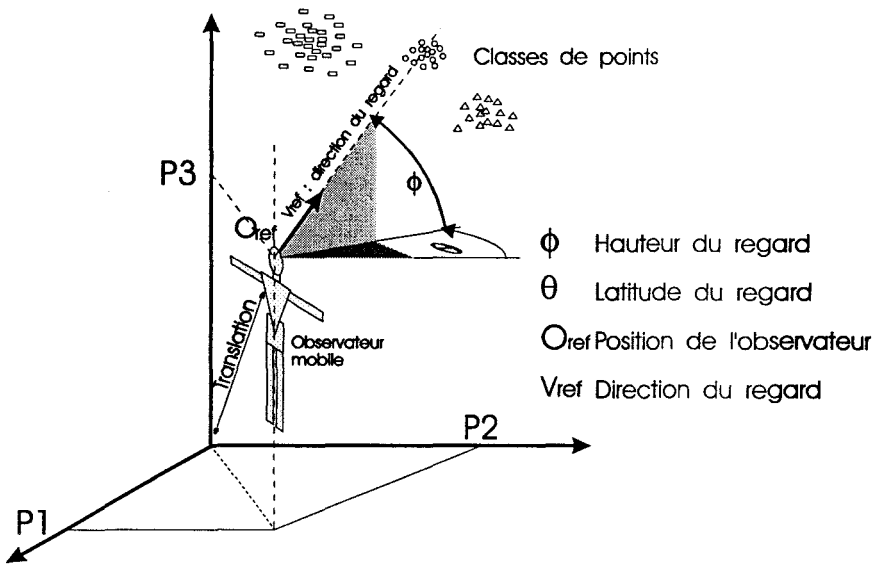


Fig. 8

Notre heuristique prend en compte un point de référence O_{ref}, position de l'observateur virtuel en déplacement dans l'espace multidimensionnel des observations, et une direction $\vec{V}_{ref}$, direction du regard de ce dernier. La représentation plane d'une observation multidimensionnelle X_q, sera donc un point $\Gamma(\rho,\alpha)$, dont les coordonnées ρ et α sont calculées par rapport à un référentiel Position/Orientation $\left(O_{ref},\vec{V}_{ref}\right)$.

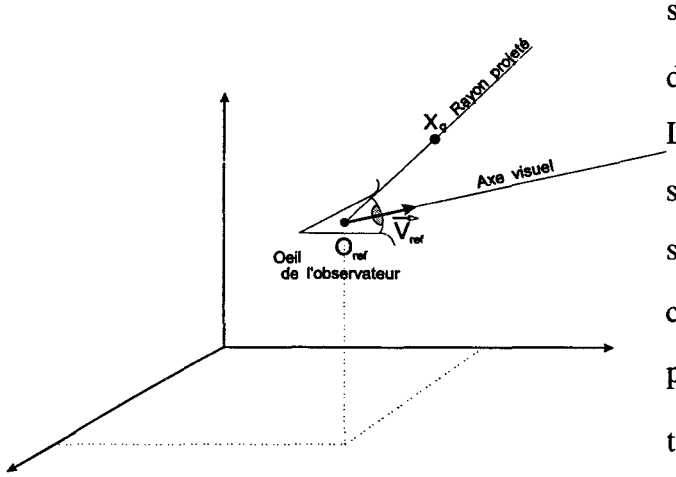


Fig. 9 : Visualisation du repère de référence

Appelons O_{ref} l'origine du référentiel mobile et $\vec{V}_{ref}$, son vecteur de référence. Reste à définir les modalités de calcul des deux coordonnées ρ et α.

Nous avons choisi pour les deux coordonnées du plan de représentation une distance ρ, et un angle α. Prenons un exemple tridimensionnel afin de préciser cette transformation (Cf. Fig. 9):

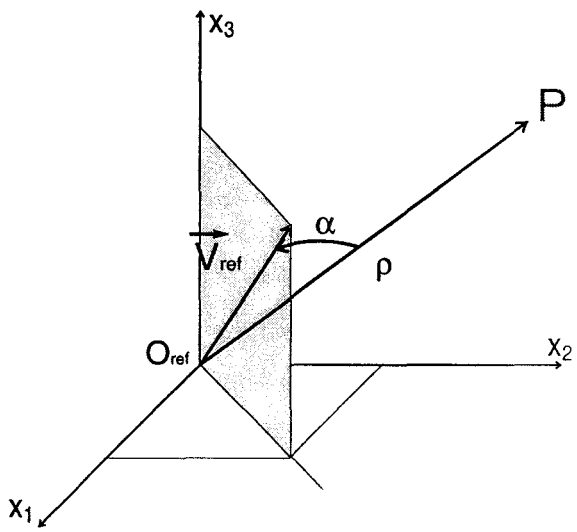


Fig. 10

Dans le plan de représentation, nous reportons en abscisse la distance Euclidienne $\rho = \left\| \overrightarrow{O_{ref} P} \right\|$ de l'observation P par rapport à l'origine du référentiel mobile O_{ref}, et en ordonnée, l'angle $\alpha = (\overrightarrow{O_{ref} P}, \vec{V}_{ref})$ (modulo π) inscrit entre le vecteur $\overrightarrow{O_{ref} P}$, et le vecteur de référence $\vec{V}_{ref}$. On aura donc:

$$\rho = \left\| \vec{O_{ref}} \vec{P} \right\| = \sqrt{\sum_{i=1}^{n} (p_i)^2} \qquad \textbf{Equ. 48}$$

$$\alpha = \arccos \frac{\left(\vec{V_{ref}} \cdot \vec{O_{ref}} \vec{P} \right)}{\rho} = \arccos \frac{\sum_{i=1}^{n} p_i . v_i}{\sqrt{\sum_{i=1}^{n} (p_i)^2}} \qquad \textbf{Equ. 49}$$

p_i : composantes de l'observation P dans l'espace d'origine,

v_i : composantes du vecteur de référence,

n : dimension de l'espace des observations.

III.2.2 *Définition du référentiel mobile*

Nous avons choisi des coordonnées qui nous paraissent les plus naturelles et aisées à manipuler:

•La position de l'origine est exprimée en coordonnées cartésiennes dans le repère original de l'ensemble des points à classer.

•Pour un espace d'origine de dimension **n**, le vecteur de référence $\vec{V}_{ref}$ peut être caractérisé par sa norme ς, et **n**-1 angles $\alpha_2, \ldots, \alpha_n$ calculés par rapport à **n**-1 axes du repère cartésien, original, en étendant à **n** dimensions les coordonnées sphériques[20] (Cf. Fig. 11).

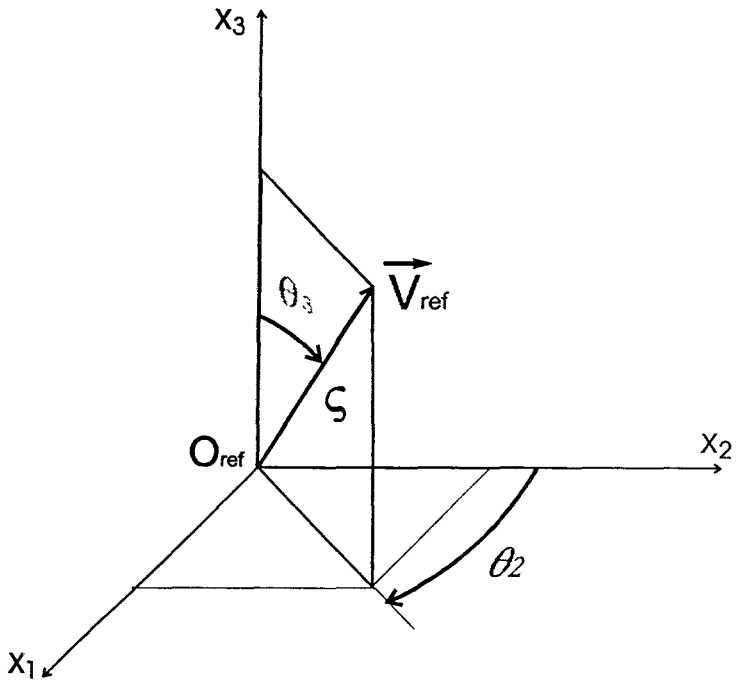


Fig. 11 : Coordonnées angulaires en dimension 3

C'est ce système de coordonnées que nous avons choisi, car le vecteur de référence servant à mesurer un décalage angulaire, seuls sa direction et son sens nous intéressent. Calculons les coordonnées cartésiennes du vecteur de référence à partir de ses coordonnées angulaires.

---

[20] Lorsque l'on fait tourner un vecteur de norme fixe autour d'un point, il semble parfaitement naturel de manipuler celui-ci à l'aide de coordonnées angulaires

Si $v_1, v_2, \ldots, v_n$ sont les coordonnées cartésiennes du vecteur $\vec{V}_{ref}$ et ς, sa norme, θ_2, ..., θ_n, les différents angles le définissant, sont ses coordonnées angulaires.

$v_1, v_2, \ldots, v_n$ pourront s'exprimer comme suit :

$$
\begin{aligned}
v_n &= \varsigma.\cos\theta_n \\
v_{n-1} &= \varsigma.\sin\theta_n \,.\, \cos\theta_{n-1} \\
v_{n-2} &= \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \cos\theta_{n-2} \\
&\quad \vdots \qquad\qquad \vdots \\
v_2 &= \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,.\, \ldots \,.\, \cos\theta_2 \\
v_1 &= \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,.\, \ldots \,.\, \sin\theta_2
\end{aligned}
\qquad \text{Equ. 50}
$$

Si $\vec{I}_1, \vec{I}_2, \vec{I}_3, \ldots, \vec{I}_n$ sont les vecteurs de la base de l'espace cartésien des observations, θ_{n-i} est l'angle que fait la projection du vecteur $\vec{V}_{ref}$ dans l'espace de dimension **n**-i défini par la base $\vec{I}_1, \vec{I}_2, \vec{I}_3, \ldots, \vec{I}_{n-i}$, avec l'axe $\vec{I}_{n-i}$, etc.. On normera le vecteur $\vec{V}_{ref}$ en imposant ς=1, pour simplifier les calculs de décalage angulaire.

Prenons l'exemple des deux classes d'observations tridimensionnelles (Cf. Fig.12).

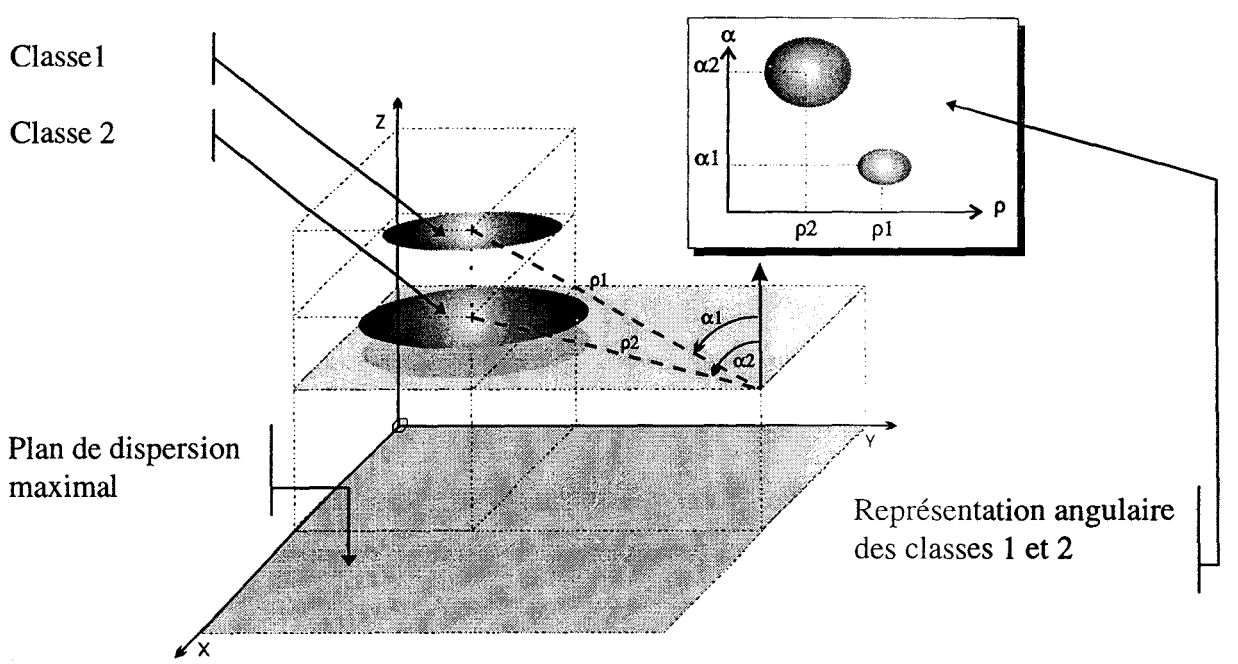


Fig. 12 : Deux classes d'observations en dim. 3

(ρ_1,α_1) et (ρ_2,α_2) sont les coordonnées des représentations angulaires des moyennes respectives des classes 1 et 2. On remarque que ρ_1 et ρ_2 sont bien distinctes ainsi que α_1 et α_2. Les représentations angulaires des deux classes d'observations apparaîtront donc de manière distincte à l'utilisateur.

Notons que l'ACP (Cf. §II.3.1.1) aurait favorisé le plan de dispersion maximal et donc les deux classes d'observations seraient apparues comme une seule dans la représentation.

La représentation basée sur ρ et α permet de bien discriminer les deux classes de points présentes.

Cette méthode de représentation présente donc à nos yeux divers avantages que nous détaillons dans le paragraphe suivant.

III.3 Avantages de l'algorithme de représentation angulaire

•L'avantage de l'algorithme de représentation angulaire le plus évident est celui de la simplicité. Les calculs nécessaires pour obtenir une représentation sont particulièrement simples, même si les observations à représenter sont nombreuses et de dimension élevée. Nous pouvons, dans l'état actuel des choses, représenter jusqu'à 16000 observations comportant, chacune, 6 paramètres[21].

•L'algorithme permet de fournir autant de représentations différentes du même ensemble de données que l'on désire, puisque le nombre de combinaisons des positions du point d'origine, ainsi que du vecteur de référence est pratiquement infini. L'utilisateur aura donc la possibilité de se faire une idée plus complète de l'ensemble des observations bien qu'il soit évident que l'être humain ne puisse construire une image mentale d'un espace comportant plus de 3 dimensions, comme il le fait avec l'espace tridimensionnel dans lequel il est plongé.

•Le calcul de chaque représentation étant particulièrement rapide, en faisant évoluer le repère de référence dynamiquement, l'utilisateur pourra, en quelque

sorte, *naviguer* à l'intérieur de l'ensemble des observations, en acquièrant ainsi une image dynamique intuitive des données. Ceci lui permettra, par exemple, de constater que tel ou tel groupe[22] de points évolue de manière similaire lorsque l'on fait varier les paramètres du référentiel mobile. Il est dès lors possible de classer les observations, non plus seulement de manière statique, mais aussi dynamique.

•De par le principe même de la représentation, un utilisateur pourra, comme dans son environnement visuel habituel, se *déplacer* vers un groupe de points en approchant l'origine, et ainsi vérifier si ce dernier ne comporte pas de *sous-groupes*. C'est le principe de granularité ou « *coarse to fine* »[23] .

•L'algorithme étant analytique, l'apport d'une nouvelle observation à l'ensemble peut être immédiatement visualisé et ne nécessite qu'un calcul propre à cette observation et elle seule, ce qui rend la méthode intéressante, entre autres, pour la surveillance de processus industriel (Cf. §II.2.1).

III.4 Etude de la représentation angulaire

III.4.1 *Validité de la représentation*

Lorsque la représentation plane comporte plusieurs groupes de points distincts, c'est-à-dire, suivant les conventions de la classification statistique, plusieurs sous-ensembles de données à densité locale forte au sein d'un ensemble à densité globale faible, on peut penser intuitivement que ces groupes existent dans l'espace d'origine. Etudions plus rigoureusement le principe de la représentation pour appuyer ce propos.

Prenons donc, pour simplifier la problématique, le cas de deux classes linéairement séparables dans le plan de représentation. Il existe alors une droite de décision permettant de distinguer les divers éléments de ces deux classes à l'aide d'un classifieur quelconque. Quel sera l'antécédent de cette droite, par

---

[21] Cette limitation est due au mode de gestion mémoire, puisque la version actuelle est compilée en 16 bits. Une version ultérieure 32 bits du programme ne devrait comporter comme limites que celles du système lui-même.
[22] nous parlons de groupes de points car aucune classification n'a encore été faite
[23] terminologie anglo-saxonne désignant une démarche partant d'une approche grossière, peu détaillée, pour aller vers une approche plus fine, plus détaillée d'un sujet

notre transformation angle/module, dans l'espace multidimensionnel des observations ?

Pour prouver que ces classes existent aussi dans l'espace multidimensionnel d'origine, il faudrait qu'il existe, entre elles, une hypersurface de séparation.

Rappelons que la représentation plane Γ, de coordonnées (ρ, α) d'un point quelconque P, de l'espace multidimensionnel $\Re^n$ d'origine, est définie par un point d'origine: O_{ref}, et un vecteur de référence $\vec{V}_{ref}$, avec :

$$O_{ref} = \begin{bmatrix} o_{ref1} \\ \vdots \\ o_{refn} \end{bmatrix} \quad \vec{V}_{ref} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \qquad \textbf{Equ. 51}$$

$\vec{V}_{ref}$ est un vecteur normé, on a donc

$$\sum_{i=1}^{n} v_i^2 = \|\vec{V}\| = 1 \qquad \textbf{Equ. 52}$$

Appelons P^* l'image du point $P = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}$ par le changement de repère $\overrightarrow{OO_{ref}}$, où O est l'origine de l'espace euclidien multidimensionnel des observations.

$$\vec{P}^* = \overrightarrow{O_{ref}P} = \begin{bmatrix} p_1^* \\ \vdots \\ p_1^* \end{bmatrix} = \begin{bmatrix} p_1 - o_{ref1} \\ \vdots \\ p_1 - o_{refn} \end{bmatrix}. \qquad \textbf{Equ. 53}$$

Les coordonnées (ρ, α) du point Γ, image plane de P par la représentation angulaire, seront alors (Equ.54, Equ.55) :

$$\rho = \|\vec{P}^*\| = \left\|\overrightarrow{O_{ref}P}\right\| = \sqrt{\sum_{i=1}^{n} (p_i^*)^2} \qquad \textbf{Equ. 54}$$

Mesure algébrique de l'angle inscrit $\longrightarrow$

$$\alpha = \overset{\wedge}{(\vec{V}_{ref}, \overrightarrow{O_{ref}P})} = \arccos\left(\frac{\sum_{i=1}^{n} v_i p_i^*}{\sqrt{\sum_{i=1}^{n} (p_i^*)^2}}\right) = \arccos\left(\frac{\vec{V}_{ref} \cdot \vec{P}^*}{\|\vec{P}^*\|}\right) \qquad \textbf{Equ. 55}$$

On obtient ainsi l'équation de la section de l'hypersurface:

$$\boxed{\arccos\frac{\alpha}{\sqrt{\alpha^2+\beta^2}} = a\sqrt{\alpha^2+\beta^2}+b} \qquad \textbf{Equ. 67}$$

Si, dans le plan $(O,\vec{V}_{ref},\vec{W})$, on utilise le système de coordonnées polaires, le point P est repéré par le couple (ρ,θ)[27], où θ est l'angle $(\vec{V}_{ref},\overrightarrow{OP})$, et ρ la norme du vecteur $\vec{OP}$, c'est à dire $\|\vec{P}\| = \sqrt{\alpha^2+\beta^2}$.

La section est donc définie en coordonnées polaires par une équation affine liant le rayon vecteur $\vec{p}^*$ et l'angle polaire θ.

L'équation de la section en coordonnées polaires est:

$$\begin{cases}\rho = C^{te} \\ 0 \le \theta \le \pi\end{cases} \qquad \textbf{Equ. 68}$$

pour des droites de décision verticales de la forme $x = C^{te}$,

et

$$\begin{cases}\theta = a\,\rho + b \\ \text{avec } 0 \le \theta \le \pi \quad \text{(Cf. §IV.4.2)} \\ \quad \text{et } \rho \ge 0\end{cases} \qquad \textbf{Equ. 69}$$

pour les autres. On note que cette courbe, lorsqu'elle n'est pas bornée, est connue sous le nom de « spirale d'Archimède ».

Mis à part les cas où, dans l'espace de représentation, la droite de décision a pour équation $y=C^{te}$ ou $x=C^{te}$, la surface ou l'hypersurface de décision[28] est

---

[27] avec les notations traditionnelles en Mathématiques

donc engendrée par la rotation d'un arc de spirale d'Archimède autour de la droite admettant pour direction le vecteur de référence $\vec{V}_{ref}$ (Cf. §IV.4.3).

Nous présentons ici un aperçu de quelques coupes de sections[29], ainsi que des surfaces tridimensionnelles engendrées, dans des cas types de segments de décisions dans le plan de représentation. Ces courbes ont été obtenues à l'aide du logiciel Matlab® 4.2.

Remarque

Dans toutes ces figures, la surface est engendrée par une rotation de la courbe le long de l'abscisse, admettant $\vec{V}_{ref}$ pour direction (Cf. Fig.15)

III.4.5 *Exemples*

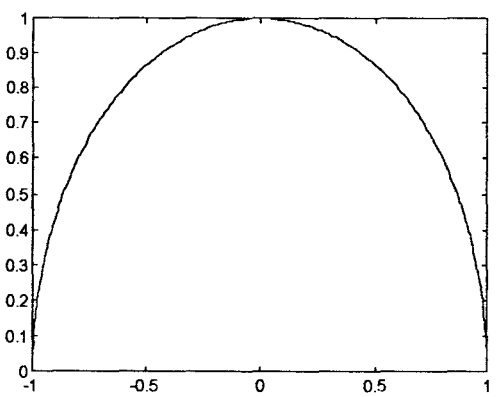
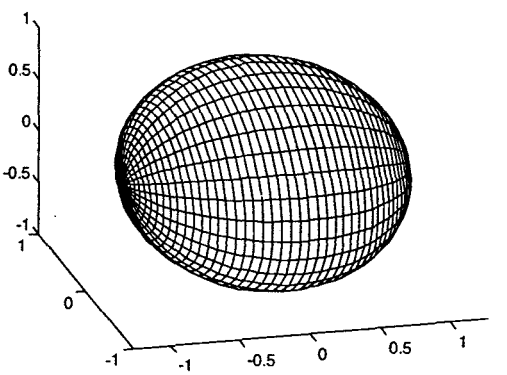


Fig. 16: $\rho = 1$



La surface engendrée est une sphère

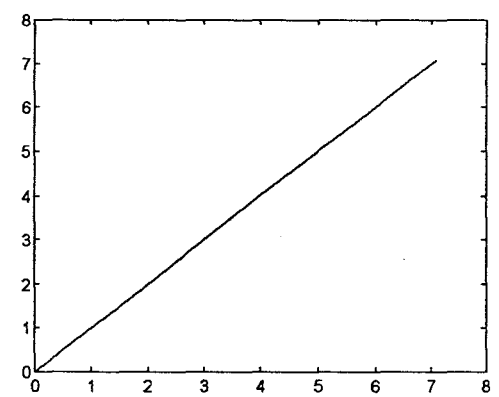
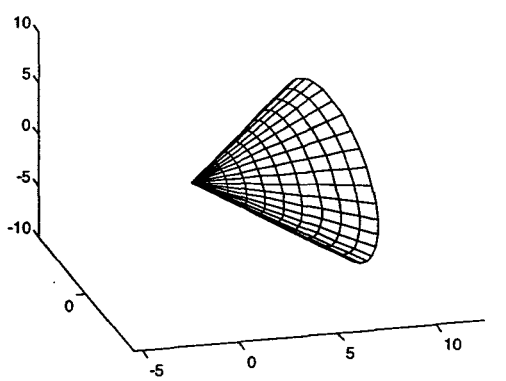


Fig. 17: $\alpha = \pi/4$



La surface engendrée est un cône

---

[29] sections de l'hypersurface de décision sections par des plans engendrés par deux vecteurs dont $\vec{V}_{ref}$ et passant par O_{ref}.

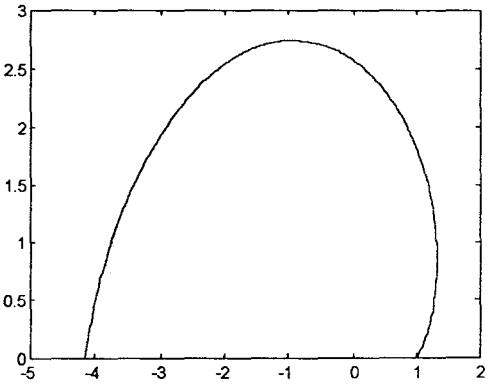
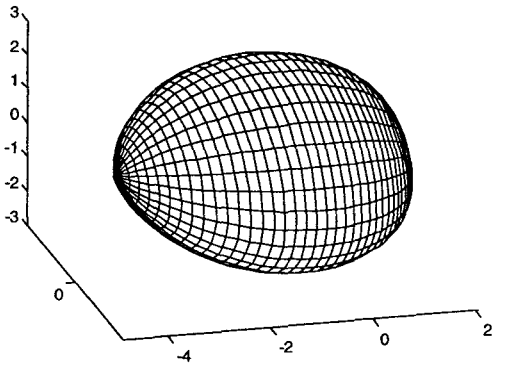


Fig. 18: $\alpha = \rho - 1$



La surface engendrée est en forme de toupie

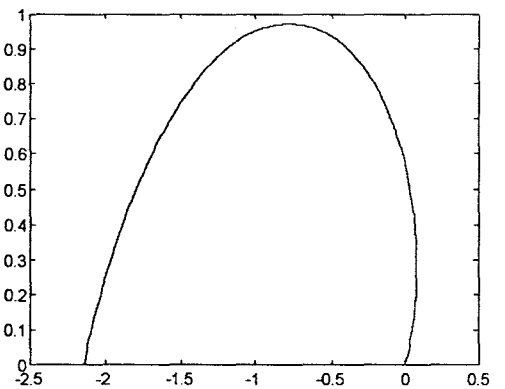
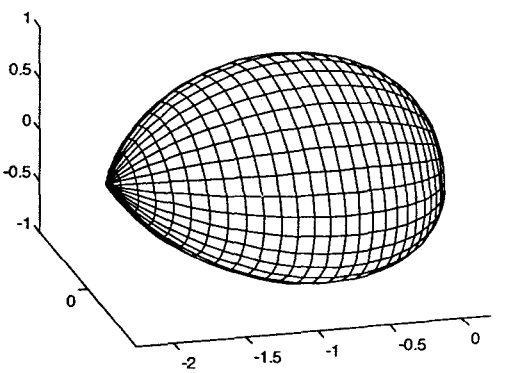


Fig. 19: $\alpha = \rho + 1$



Surface engendrée par Fig.19

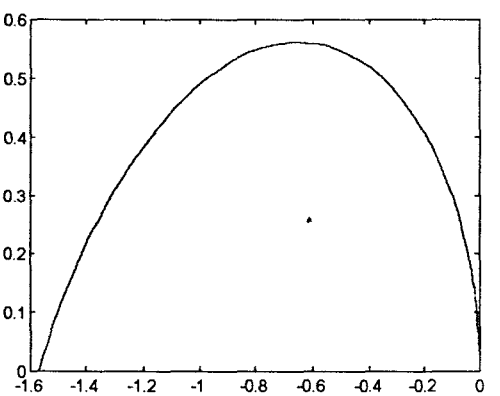
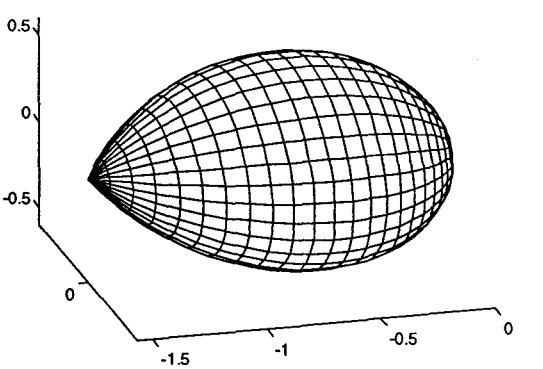


Fig. 20: $\alpha = \rho + \pi/2$



Surface engendrée par Fig. 20

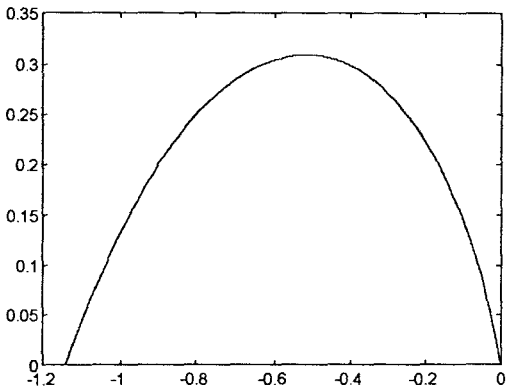
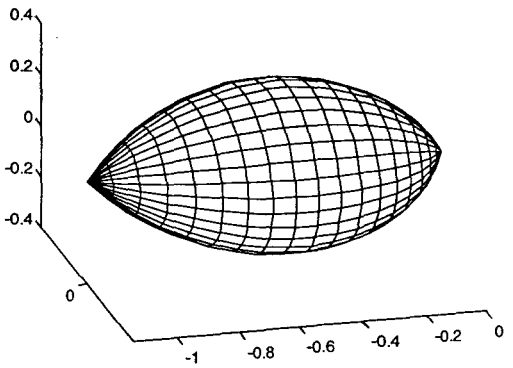


Fig. 21: $\alpha = \rho + 2$



Surface engendrée par Fig. 21

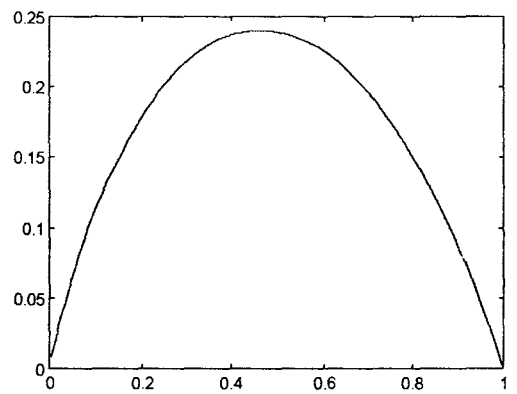
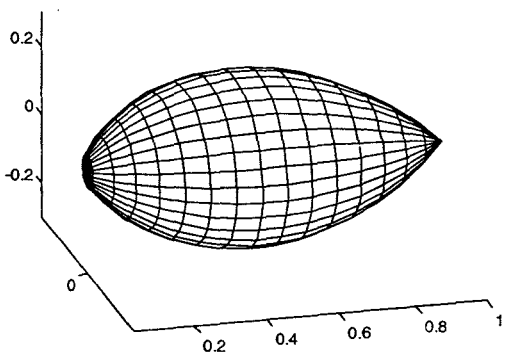


Fig. 22: $\alpha = -\rho + 1$



Surface engendrée par Fig. 22

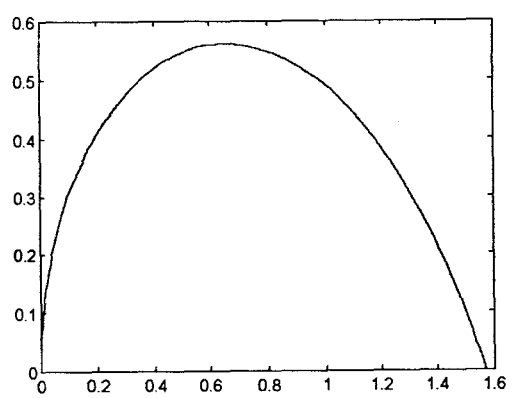
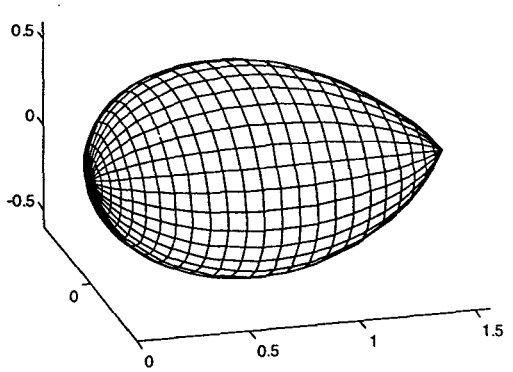


Fig. 23: $\alpha = -\rho + \pi/2$



Surface engendrée par Fig. 23

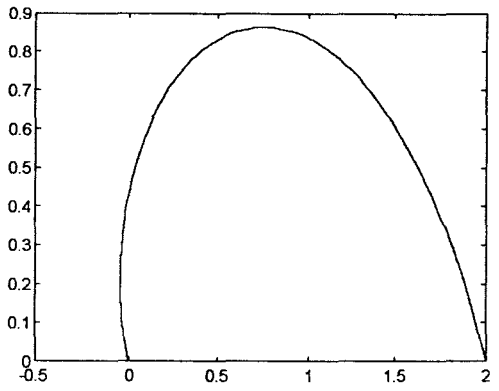
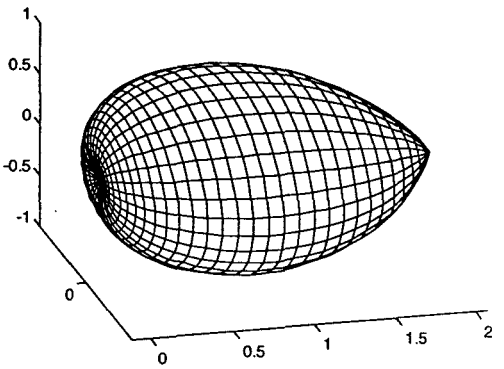


Fig. 24: $\alpha = -\rho + 2$



Surface engendrée par Fig. 24

III.4.6 *Conclusion*

Dans ce chapitre, nous avons passé en revue les aspects les plus significatifs de notre algorithme de représentation. Il s'avère, après cette étude, que cet algorithme est valide, et qu'il présente des potentialités très intéressantes. Mais, contrairement à de multiples algorithmes de classification, une simple implantation sur calculateur ne permettrait pas d'exploiter de manière exhaustive ses capacités. En effet, lorsque nous avons envisagé cet algorithme, c'était dans le but qu'un utilisateur lambda, ne possédant pas forcément de connaissances en analyse de données et ne disposant pas d'un système particulièrement performant et onéreux, puisse aborder la classification d'un ensemble d'observations de manière simple, rapide et intuitive, par le biais de notre algorithme.

Pour cela, il est primordial de concevoir une interface utilisateur aussi agréable et simple que possible à utiliser.

C'est ce sujet que nous traitons au chapitre suivant où nous abordons quelques points importants d'ergonomie en matière de conception de logiciels.

CHAPITRE IV
LES IMPLANTATIONS DE LA REPRESENTATION ANGULAIRE

IV. LES IMPLANTATIONS DE LA REPRESENTATION ANGULAIRE

IV.1 Généralités

☞ Choix du système

Il est aisé d'expliquer le choix du PC pour le développement de notre application : en effet, l'importance de l'implantation de ce dernier dans tous les secteurs de notre société est bien connue, et nous voulions créer une application qui soit accessible au plus grand nombre, et non plus réservée aux utilisateurs universitaire de gros systèmes, comme c'est souvent le cas en traitement de données.
Pour exploiter pleinement le type de représentation plane que nous avons choisi, l'interface graphique s'impose[JMG95] : il faut permettre à l'opérateur de manipuler le référentiel mobile (à savoir le point d'origine et la direction du vecteur de référence) de la manière la plus simple et naturelle possible. L'utilisateur doit pouvoir se concentrer sur les données qu'il étudie sans avoir à se préoccuper du logiciel qu'il utilise : c'est le principe d'ergonomie de transparence bien connue des concepteurs d'interfaces graphiques.

Nous nous sommes donc attachés à ce que le logiciel conçu soit le plus *transparent* et ergonomique possible, par un choix adéquat de la manière dont les informations sont affichées et les commandes saisies.

☞ Choix du langage

Pour implanter notre concept, nous avons choisi le langage C, et ceci pour deux raisons :
- Le langage C est connu pour sa rapidité d'exécution, et la compacité des programmes créés à l'aide de celui-ci sur de petits systèmes. C'est un aspect primordial, étant donnée la masse très souvent importante des données à analyser.
- Le langage C est proche du langage machine sous bien des aspects, et il présente une grande puissance pour le traitement des pointeurs, ce qui, nous le verrons plus tard, nous intéresse tout particulièrement pour la gestion de bases de données importantes. De plus, la plupart des logiciels actuels sont écrits en C ou en C++ (orienté objet)

☞ Choix du support graphique

☞ Un prototype a été implanté à l'aide du logiciel utilitaire de programmation **METAWINDOW™**.

* **METAWINDOW™** fournit une base de procédures permettant à la fois le multi-fenêtrage, l'utilisation de la souris, et les tests d'inclusion de points dans des figures pré-définies, dans un programme tournant sous DOS. Cet environnement simple suffisait donc amplement pour une première étude de notre principe de représentation angulaire

☞ Le logiciel définitif a été conçu en langage C sous Windows.

* **MS Windows** fournit en effet un environnement graphique parfaitement adapté à l'implantation et aux extensions ultérieures de notre application. L'environnement Windows permet, en application des principes de base d'ergonomie, une utilisation transparente et intuitive des logiciels, tout en fournissant de surplus une palette de fonctionnalités et d'objets préexistants permettant l'écriture d'un logiciel parfaitement ergonomique.

☞ Cependant, nous avions tout d'abord besoin de pouvoir mettre en oeuvre la plupart des idées de bases du logiciel projeté, sans la lourdeur d'apprentissage qu'impliquait une implantation sous MS Windows afin de tester la validité de nos idées sur un prototype plus simple.

IV.2 Implantation sous Metawindows : le programme INTERACT

IV.2.*1* *Le retour d'information*

IV.2.*1*.1 Mode d'affichage des coordonnées des point et vecteur de référence

Afin de fournir à l'utilisateur un retour d'information « visuel » sur l'état du référentiel, nous avions d'abord pensé visualiser les positions respectives de l'origine du référentiel et du vecteur de référence à l'aide de diagrammes de type « toiles d'araignées » (Cf. Fig. 25).

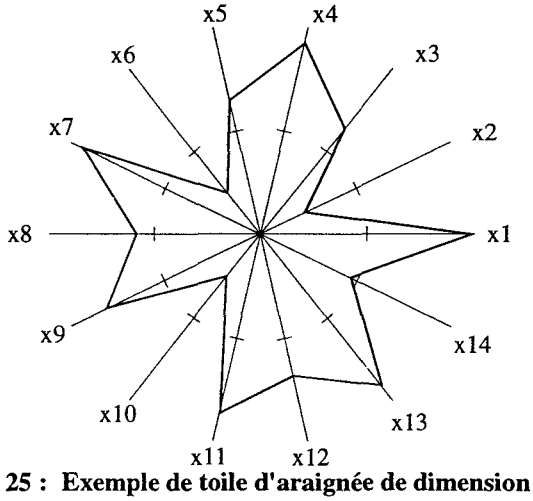


Fig. 25 : Exemple de toile d'araignée de dimension 14

Ce type de diagramme, souvent utilisé lorsque la dimension de l'espace des paramètres dépasse trois, présente l'avantage d'être très *parlant* pour l'utilisateur. Ce dernier peut aisément retenir visuellement une configuration des paramètres définissant le référentiel mobile dans l'espace multidimensionnel.

Cependant, l'aspect angulaire de la position du vecteur référence s'est révélé mal traduit. De plus, étant donnée la taille de l'écran ainsi que l'emplacement que l'on devait réserver à la représentation des observations, ce type d'affichage serait vite devenu illisible pour un nombre important de paramètres. C'est pourquoi nous avons finalement opté pour un type d'affichage plus simple et plus facile à manier :

- La position de l'origine est représentée par des curseurs se déplaçant sur des *barrettes* (Cf. Fig. 26-❸ et 27).

- Pour représenter de manière la plus imagée possible la position du vecteur de référence, en coordonnées angulaires généralisées, il fallait opter pour un affichage faisant intervenir un angle. C'est pourquoi cette position est affichée à l'aide de

portions de disques ou *camemberts* (Cf. Fig. 26-❷ et 27). Si la dimension de l'espace des observations est **n**, les *camemberts* seront au nombre de **n-1** (Cf. §III.2.2)

Les barrettes et les portions de disques sont de couleurs différentes pour chaque axe de référence, permettant ainsi une meilleure lisibilité pour l'opérateur.

L'interface d'INTERACT se présente comme schématisée sur la Figure 26.

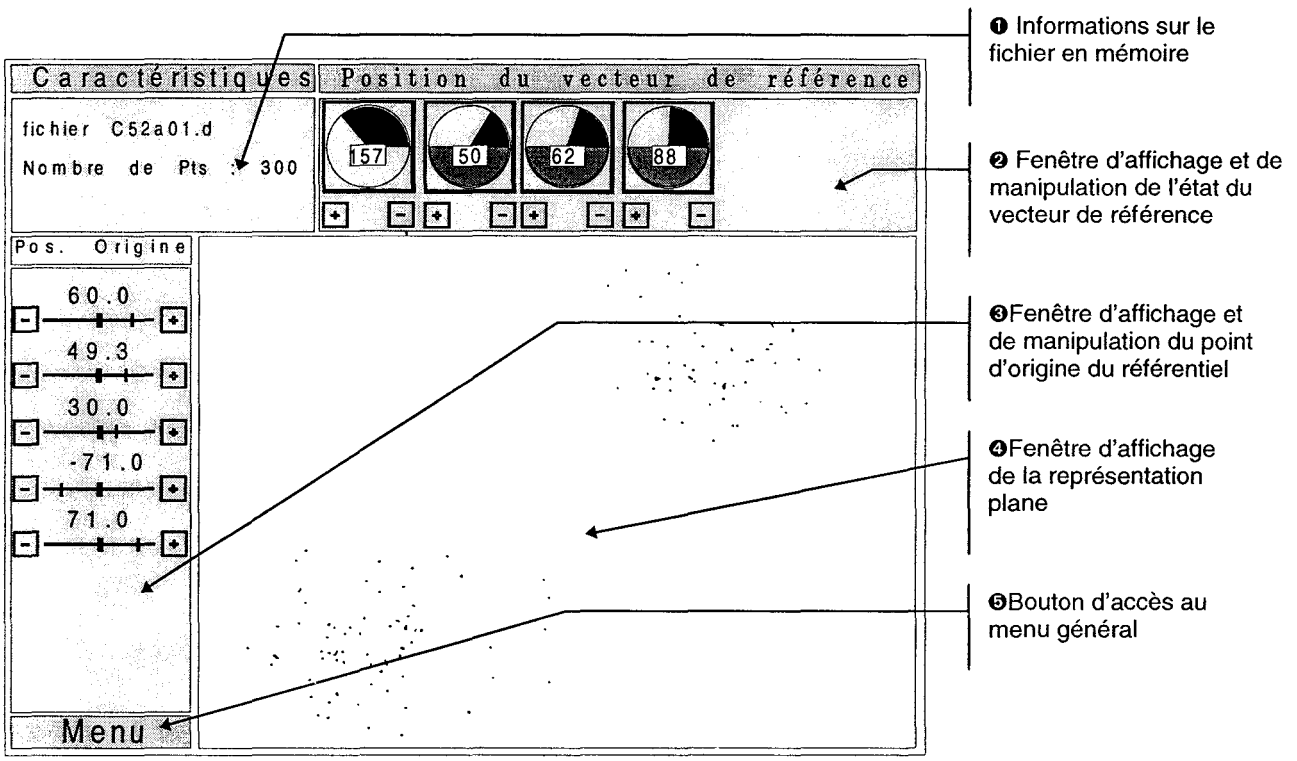


Fig. 26 : Interface Graphique de l'application sous Metawindows

Ces modes d'affichage des positions respectives de l'origine et du vecteur de référence possèdent, par rapport aux diagrammes évoqués précédemment, l'avantage de pouvoir être utilisés pour des espaces de dimension importante, tout en gardant leur lisibilité. Cependant, bien que ces deux modes d'affichage permettent de bien percevoir les modifications opérées sur le référentiel, ils ne donnent pas de renseignements assez précis sur les positions respectives de l'origine de ce dernier, ni du vecteur de référence. C'est pourquoi, toujours dans le souci du retour d'information vers l'utilisateur, nous avons aussi affiché au niveau de chaque *barrette* et de chaque *camembert*, les valeurs numériques de la variable correspondante.

IV.2.1.2 Affichage des valeurs numériques

♦ Origine

La latitude L_i de déplacement de l'origine, sur l'axe i, est la fourchette des valeurs que peut prendre la coordonnée d'ordre i de l'origine de notre référentiel. Celle-ci est fixée de la manière suivante:

$$L_i = \max(|x_i \max|, |x_i \min|)$$

où x_i **max**, et x_i **min** sont respectivement les valeurs maximale et minimale signées du $i^{ème}$ paramètre (x_i min peut être négatif et supérieur en valeur absolue à x_i max).

Pour afficher la valeur numérique du décalage de l'origine, nous avons cru bon d'effectuer une normalisation (à l'affichage uniquement), afin que l'utilisateur puisse effectuer des comparaisons entre les différents paramètres.

Cette normalisation est effectuée en multipliant la valeur du décalage par un facteur d'échelle E_i avec : $E_i = 100 / L_i$ (Cf. Fig. 28)

♦ Vecteur de référence

La valeur, en degrés, de chaque coordonnée angulaire du vecteur de référence est affichée en surimpression sur le *camembert* correspondant. (Cf. Fig. 27)

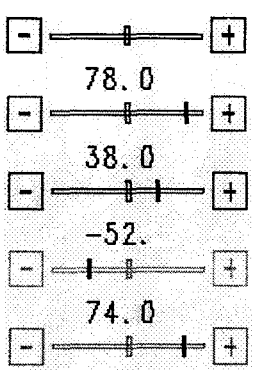


Fig. 28: Origine

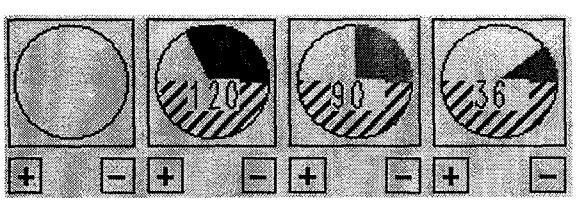


Fig. 27: Vecteur de référence

IV.2.1.3 Visualisation des points de la représentation plane

Après l'évaluation, par une procédure de comparaison, des extrema en abscisse et en ordonnée, de l'ensemble des points images des observations, l'affichage est cadré à partir de ces extrema, de manière à ce que l'ensemble des pixels images obtenus emplisse la fenêtre de représentation (Cf. Fig.29) ; i.e.:

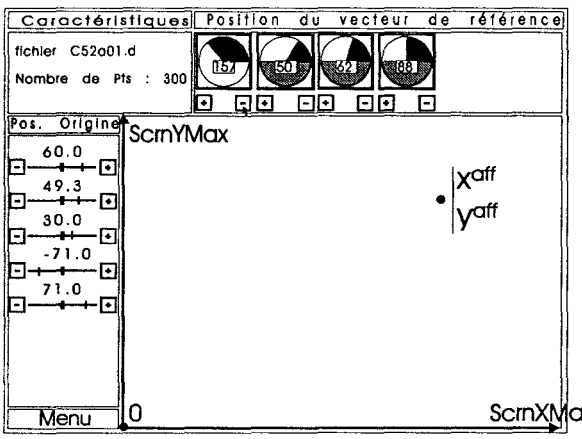


Fig. 29 : Zone Ecran de visualisation des points images

- $x^{aff} = (x - xmin) . \frac{ScrnXMax}{xMax - xmin}$
-
- $y^{aff} = (y - ymin) . \frac{ScrnYMax}{yMax - ymin}$

avec :

- ScrnXMax, ScrnYMax : maxima des coordonnées de l'écran (les minima étant nuls),
- x^{aff}, y^{aff} : coordonnées du pixel image (pixel écran correspondant au point image d'une observation par la transformation angulaire)
- x , y : coordonnées du point image d'une observation par la transformation angulaire.
- xmin, xMax, ymin, yMax : extrema des points images des observations.

par la transformation angulaire.

IV.2.*1*.4 Affichage des paramètres d'une observation

Lors des premiers essais du logiciel, nous déplacions l'origine du référentiel en jugeant de la qualité du décalage uniquement sur l'apparence de la représentation des données. Ce type de modification du référentiel s'est vite avéré insuffisant. C'est pourquoi nous avons ajouté une aide au déplacement de l'origine, en permettant au praticien d'afficher les coordonnées d'un point désigné avec la souris. Cette possibilité, qui permet à l'analyste de connaître les coordonnées de l'observation

multidimensionnelle associée au pixel image sélectionné, résulte de l'exploitation de la structure particulière des fichiers.

L'opérateur indique l'emplacement du pixel image choisi avec le bouton gauche de la souris, puis valide sa demande avec le bouton droit. Les coordonnées normalisées de l'observation multidimensionnelle associée (Cf. §V.2.1.2) sont alors affichées à l'emplacement qu'occupe le curseur lors de la validation (Cf. Fig. 30).

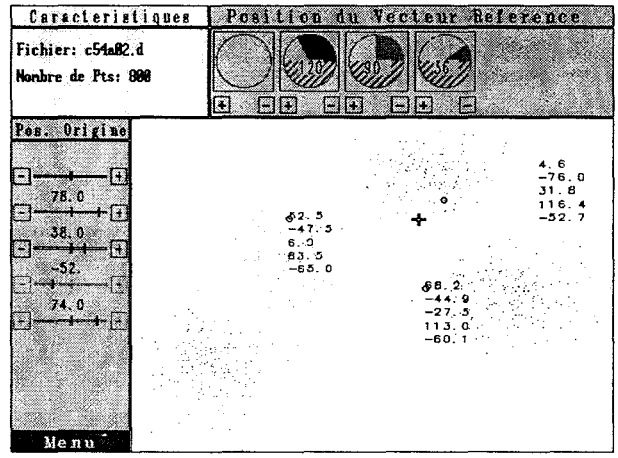


Fig. 30: exemple d'affichage des paramètres de trois observations

IV.2.2 *Les fonctions de l'application INTERACT*

Le prototype INTERACT sert à valider notre concept de représentation plane afin de prouver sa faisabilité.

Les fonctions qui ont été implantées sont donc des fonctions permettant l'affichage de la représentation Module/Angle, ainsi que les fonctions de base permettant la classification à partir d'une ou de plusieurs représentations d'un même ensemble d'observations[ESS93].

IV.2.2.1 Saisie des fichiers d'observations

Cette manoeuvre est réduite au strict minimum dans le cas de notre prototype. La saisie d'un fichier initial se fait automatiquement au lancement du programme. Pour effectuer la saisie d'un nouveau fichier en cours de programme, il suffit de sélectionner le menu général (Fig. 26-❺, Fig. 31), avec la souris, et de la même manière, de sélectionner dans ce menu la saisie d'un nouveau fichier.

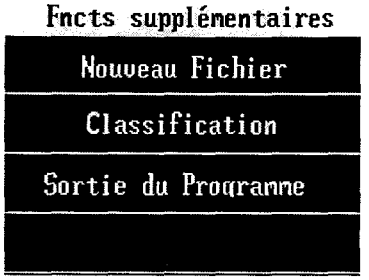


Fig. 31: Menu général

IV.2.2.2 Modification de l'état du référentiel / Affichage des points.

L'opérateur modifie la position du vecteur de référence, ou de l'origine, en *cliquant* avec la souris dans les carrés d'incrémentation ou de décrémentation qui sont placés au niveau de l'affichage des grandeurs correspondantes (Cf. Fig. 26-❷, 26-❸, 27 et 28).

On peut déplacer l'origine de $-L_i$ à $+L_i$ sur l'axe I_i (Cf. §V.2.1.2), et les coordonnées angulaires du vecteur de référence sont toutes comprises entre 0° et 180°, sauf le premier angle qui est compris entre 0° et 360°, ceci, à l'instar des coordonnées sphériques, pour éviter les redondances.

Il y a deux modes de modification du référentiel :

- En mode *pas à pas*, avec le bouton de gauche de la souris, à chaque modification du référentiel, l'indicateur de position correspondant est remis à jour, la représentation Module/Angle est recalculée, et l'affichage rafraîchi.
- En mode *différé* avec le bouton de droite de la souris, tant que le bouton est enfoncé, nous incrémentons ou décrémentons la grandeur sélectionnée (avec un pas plus petit qu'en vitesse lente). Seul l'indicateur de position correspondant évolue. Les calculs, ainsi que le rafraîchissement de la représentation plane, ne sont effectués qu'après relâchement du bouton droit de la souris. (Cette formule s'est avéré nécessaire à cause de l'augmentation du temps de calcul de la représentation pour des bases de données importantes)

IV.2.2.3 Fonctions de classification interactive

Le logiciel comprend quatre fonctions de gestion de classes qui sont activées par le choix du bloc de classification dans le menu général. On active les procédures de classification en *cliquant* avec la souris dans la case correspondante du menu

général(Fig. 32). Les fonctions de modification du référentiel mobile sont alors invalidées et la fenêtre « routines de classification » remplace alors la fenêtre de manipulation du vecteur de référence (Cf. Fig. 32

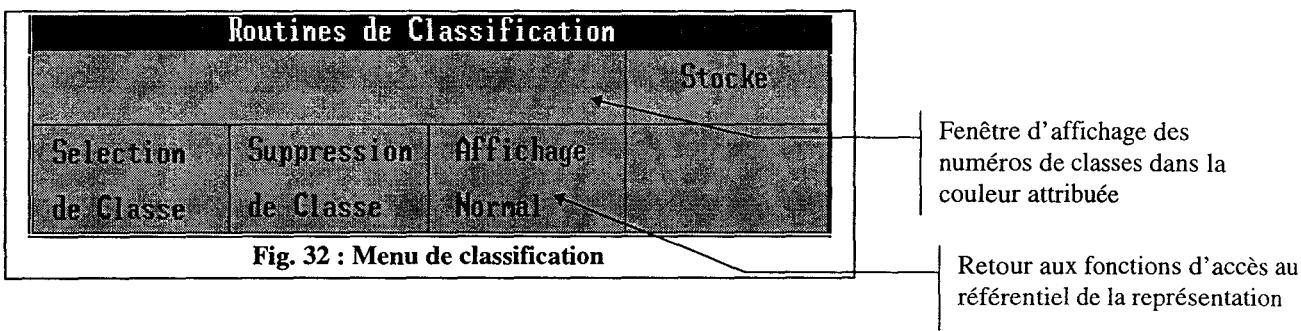


Fig. 32 : Menu de classification

☞ Pour des raisons de simplicité, la création des classes d'observations se fait en entourant les pixels images sélectionnés avec un polygone comprenant 20 sommets au maximum, créé à l'aide de la souris : l'utilisateur choisit l'emplacement du sommet à l'aide du bouton de gauche, puis valide l'emplacement choisi avec le bouton de droite. Enfin, il indique le dernier sommet, en pressant simultanément sur les deux boutons de la souris (Cf. Fig.33).

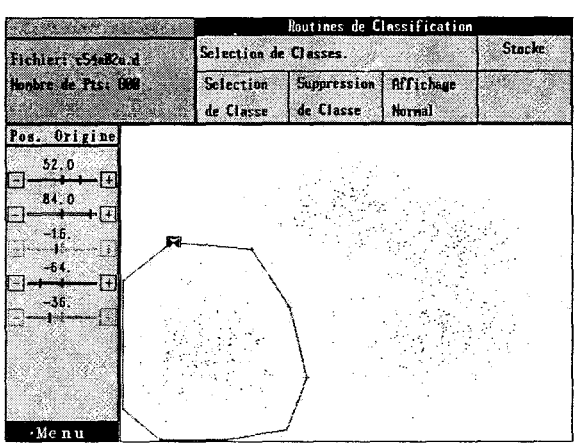


Fig. 33 : Création d'une classe d'observations

Un test d'inclusion des pixels images dans la figure ainsi créée est alors effectué. Chaque fois qu'un pixel image situé à l'intérieur du polygone est détecté, l'observation dont il est la représentation est assignée à la classe en cours de création, et déplacée dans une zone de mémoire associée à cette classe, et ainsi de suite jusqu'au dernier pixel image de l'ensemble des observations.

La procédure de réorganisation des classes attribue alors aux points sélectionnés une couleur spécifique, et les réaffiche dans cette couleur. Le numéro de la classe, et le nombre de points qu'elle contient sont affichés

Chaque fois qu'une nouvelle classe est créée, la base de données en mémoire vive est restructurée. Si par exemple, l'opérateur a créé deux classes d'observations, la base de données sera structurée en trois ensembles : les deux classes créées seront en tête, et la dernière « classe », sera en fait, l'ensemble des points non classés.

☞ La procédure de suppression de classe permet à l'utilisateur de refondre une classe d'observations qu'il a créée avec l'ensemble des observations non classées (Cf. Fig.33). Cette procédure ne peut bien sûr être lancée que si le nombre de classes est supérieur à 1, la dernière *classe* étant, par convention, le groupe des observations non classées, et ne pouvant être supprimée.

Lorsque la suppression de classe est activée, la procédure affiche le numéro de chaque classe existante, dans la couleur qui lui a été attribuée. Il suffit d'entrer au clavier le numéro de la classe à supprimer. Les observations appartenant à cette classe, sont alors assignées à la dernière *classe* des points non classés, et les pixels images correspondants sont réaffichés dans la couleur du groupe des observations non classées.

☞ Lorsque l'on est satisfait d'une classification ainsi effectuée, on peut stocker les observations reclassées, dans un fichier en mémoire de masse. Ce fichier est indépendant du fichier original des observations, et porte le même nom que ce dernier, avec une extension différente : l'extension " .cla".

☞ Sécurités

Le mode de classification choisi interdit de placer dans une nouvelle classe un point déjà classé. Lorsque l'on veut stocker en mémoire de masse une nouvelle classification, et qu'un classement y existait déjà, une fenêtre est affichée, demandant de confirmer la requête. Si celle-ci est confirmée, le fichier existant est écrasé par le nouveau fichier créé.

IV.2.3 *Quelques détails de Programmation*

IV.2.3.1 Calcul des coordonnées cartésiennes du vecteur de référence

On rappelle que la position du vecteur de référence est fixée par le biais de ses coordonnées angulaires généralisées, mais que tous les calculs sont effectués en coordonnées cartésiennes, d'où la nécessité d'une conversion.

On a vu que si $v_1,\ldots, v_n$ sont les coordonnées cartésiennes du vecteur de référence $\vec{V}_{ref}$, et ς, θ_2, ..., θ_n ses coordonnées angulaires généralisées,

$v_1,\ldots, v_n$ peuvent s'exprimer comme suit :

$$v_n = \varsigma.\cos\theta_n$$
$$v_{n-1} = \varsigma.\sin\theta_n \,.\, \cos\theta_{n-1}$$
$$v_{n-2} = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \cos\theta_{n-2}$$
$$\text{"} \quad \text{"}$$
$$\text{"} \quad \text{"}$$
$$v_2 = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,.\,.\, \cos\theta_2$$
$$v_1 = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,.\,.\, \sin\theta_2$$

- $\vec{I}_1,\vec{I}_2,\vec{I}_3,\ldots,\vec{I}_n$ sont les vecteurs de la base de l'espace cartésien des observations.

- θ_{n-i} est l'angle que fait la projection du vecteur $\vec{V}_{ref}$ dans l'espace de dimension n-i défini, par les axes $\vec{I}_1,\vec{I}_2,\vec{I}_3,\ldots,\vec{I}_{n-i}$, avec l'axe $\vec{I}_{n-i}$. (Cf. §III.2.2)

On remarque ici l'aspect répétitif de ces formules. Nous nous sommes donc appliqués à trouver une programmation récurrente pour le calcul des coordonnées cartésiennes du vecteur de référence $\vec{V}_{ref}$ à partir de ses coordonnées angulaires, sa norme étant, par principe, fixée à 1.

☞ On peut construire les coordonnées cartésiennes du vecteur de référence, à partir des coordonnées angulaires généralisées, en incrémentant la dimension de l'espace par étapes. Le calcul sera effectué comme suit :

1ère étape : Coordonnées polaires dans le sous espace de base I_1, I_2

$$v_2 = \varsigma \cdot \cos \theta_2$$
$$v_1 = \varsigma \cdot \sin \theta_2$$

2ième étape : Coordonnées sphériques dans le sous espace de base I_1,I_2, I_3

$$v_3 = \varsigma \cdot \cos \theta_3$$
$$v_2 = \varsigma \cdot \sin \theta_3 \cdot \cos \theta_2$$
$$v_1 = \varsigma \cdot \sin \theta_3 \cdot \sin \theta_2$$

.
.

jième étape : Dimension j+1, sous espace de base I_1, I_2,... , I_{j+1}

$$v_{j+1} = \varsigma.\cos \theta_{j+1}$$
$$v_j = \varsigma.\sin \theta_{j+1} \cdot \cos \theta_j$$
" "
" "
$$v_2 = \varsigma.\sin \theta_{j+1} \cdot \sin \theta_j \cdot \sin \theta_{j-1} \cdot /.../ \cdot \cos \theta_2$$
$$v_1 = \varsigma.\sin \theta_{j+1} \cdot \sin \theta_j \cdot \sin \theta_{j-1} \cdot /.../ \cdot \sin \theta_2$$
.
.

Etape finale : Dimension ***n***, espace d'origine, de base I_1, I_2, ... , I_n

$$v_n = \varsigma.\cos \theta_n$$
$$v_{n-1} = \varsigma.\sin \theta_n \cdot \cos \theta_{n-1}$$
$$v_{n-2} = \varsigma.\sin \theta_n \cdot \sin \theta_{n-1} \cdot \cos \theta_{n-2}$$
" "
" "
$$v_2 = \varsigma.\sin \theta_n \cdot \sin \theta_{n-1} \cdot \sin \theta_{n-2} \cdot /.../ \cdot \cos \theta_2$$
$$v_1 = \varsigma.\sin \theta_n \cdot \sin \theta_{n-1} \cdot \sin \theta_{n-2} \cdot /.../ \cdot \sin \theta_2$$

L'algorithme de calcul correspondant sera donc le suivant (Fig. 35):

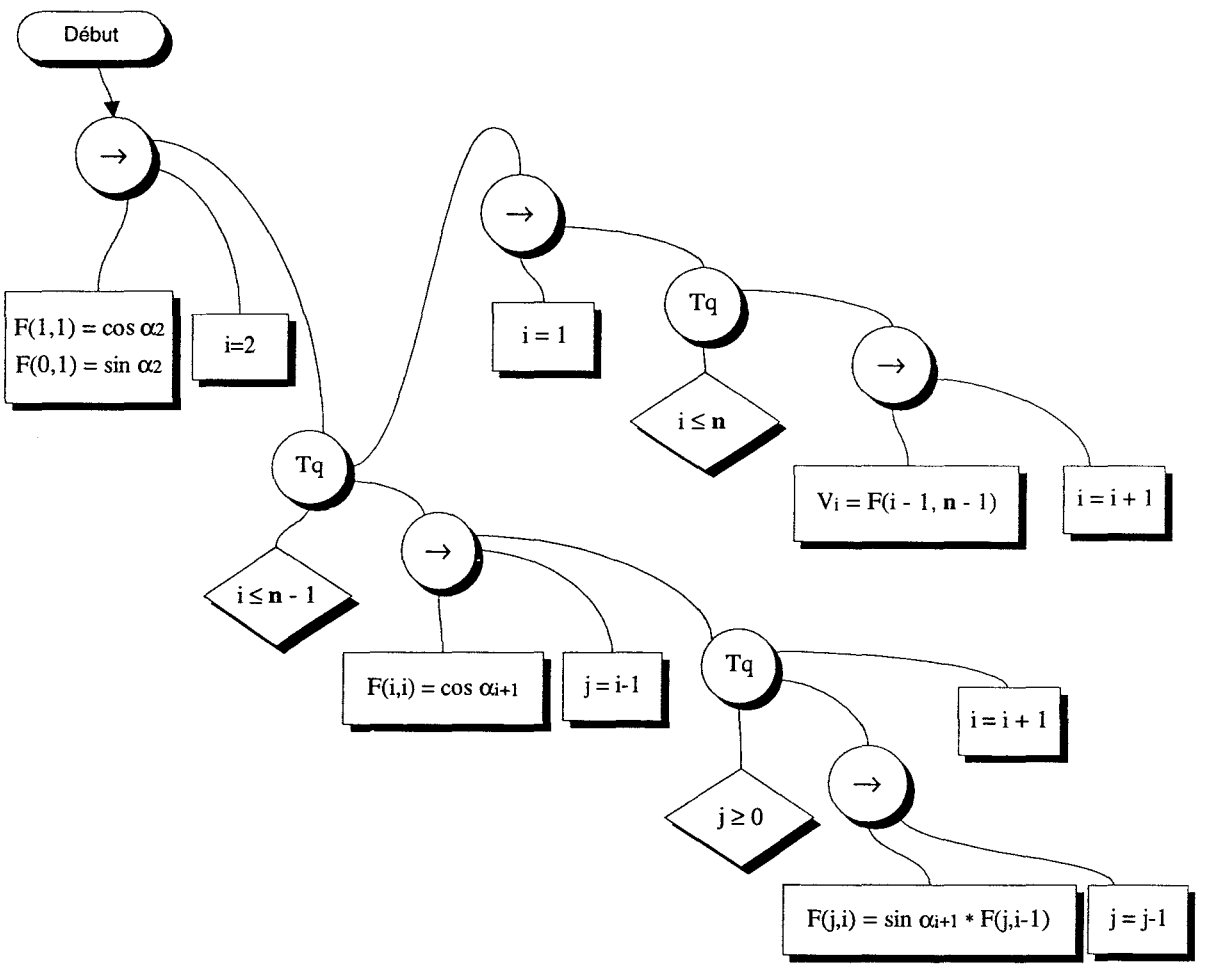


Fig. 34 Arbre programmatique de calcul des coordonnées du vecteur de référence

IV.2.3.2 Organisation des données

IV.2.3.2.a)*Structure des fichiers de données en mémoire de masse*

La structure de base des fichiers de données, ainsi que sa forme, reprend celle utilisée actuellement dans l'équipe de chercheurs « image et décision » du Centre d'Automatique de Lille. Cette structure a l'avantage d'être transportable aisément, puisque les données sont stockées sous la forme de caractères ASCII. Elle permet par ailleurs une grande souplesse d'utilisation, puisque les bases de données classées et non classées sont structurées de la même manière.

Les données apparaissent comme suit:

$\mathbf{p}, \mathbf{n}, \mathbf{K}$

$\mathbf{p}_1, \ldots, \mathbf{p}_K$

$x_1(1), \ldots , x_n(1)$
$\vdots \quad \vdots$
$x_1(\mathbf{p}_1), \ldots , x_n(\mathbf{p}_1)$
$\vdots \quad \vdots$
$x_1(\mathbf{p}_1 + \mathbf{p}_2), \ldots , x_n(\mathbf{p}_1 + \mathbf{p}_2)$
$\vdots \quad \vdots$
$x_1(\mathbf{p}), \ldots , x_n(\mathbf{p})$

Avec **p** : nombre total d'observations, ***n*** : dimension de l'espace des paramètres, **K** : nombre de classes (dans le cas d'un fichier d'observations non classées, **K** = 1), $\mathbf{p}_1, \ldots, \mathbf{p}_K$: nombre d'observations par classe, $x_j(i)$: paramètre d'ordre j de la $i^{ième}$ observation.

☞ **Structure des données en mémoire vive**

En mémoire vive, les données sont aussi stockées sous une forme structurée, à l'aide de pointeurs. Les observations et leurs représentations planes sous forme de points, ont le même indice, ce qui permet de retrouver aisément l'observation originale à partir de sa représentation.

Il n'est pas rare d'avoir à traiter des bases de données de plusieurs centaines observations, dans un espace de plus de 3 dimensions. Les observations sont stockées en mémoire dans des variables de type réel double précision, donc pour un tableau de 1500 observations en 7 dimensions, l'occupation mémoire sera de 82 K octets. Or le langage C sous système DOS ne peut pas gérer, avec 8 bits d'adressage, des tableaux de plus de 64 K octets chacun.

C'est pourquoi nous avons choisi de stocker ces données en mémoire vive, sous forme de blocs structurés adressés par pointeurs. Nous aurons 4 blocs organisés de la sorte :

- Le bloc des observations originales,
- Le bloc des observations, après le changement d'origine de l'espace multidimensionnel (Cf. Chap. III),
- Le bloc de la transformée plane des observations,
- Le bloc des pixels image résultants.

Les données des deux premiers blocs sont ordonnées de la même manière que dans le fichier original des observations en mémoire de masse.

Ce mode de stockage a présenté un avantage très net de souplesse par rapport au stockage dans des tableaux de format fixe, lors de la conception des procédures de classification interactive. En effet :

- Bien qu'intrinsèquement, le C ne fasse pas de différence réelle entre tableaux et pointeurs, la gestion directe par pointeurs s'est révélée plus efficace et rapide, avec le compilateur C dont nous disposions, que celle par tableaux. Il en a résulté un gain de temps d'un facteur 10 environs dans le reclassement des points lors de la construction des classes, ainsi que dans le rafraîchissement de la représentation.

- Lors de la suppression d'une classe, ce mode de mémorisation nous a permis de déplacer les données par blocs (opérations de bas niveau mémoire), permettant d'effectuer l'opération de déplacement de la classe supprimée dans le groupe des points non classés, ainsi que le décalage des autres classes, en quelques dixièmes de secondes, alors que la même opération effectuée point par point prenait de une à deux minutes.

IV.2.3.3 Extensions

Nous avons complété le fichier des observations décrit précédemment, par un autre fichier qui porte le même nom, mais auquel nous avons affecté une extension différente : Un fichier « .cla », généré par les fonctions de classification, contient l'ensemble des observations après classification. Ce fichier a, bien entendu, la même structure que le fichier original.

IV.2.4 *Limitations d'INTERACT*

Les limitations de notre prototype sont de trois ordres :

IV.2.4.1 Limitations d'affichage

- Limitations dues au format d'affichage du standard VGA 16 couleurs géré par METAWINDOW™ qui ne comporte que **16** couleurs affichables dans une palette de 64 couleurs (les couleurs sont définies sur un octet). Le nombre effectif de couleurs disponibles est donc de 16, auquel il faut retrancher les couleurs réservées pour les indicateurs de position du référentiel. Nous sommes donc limités à 10 classes d'observations maximum.

- Limitations dues à la taille de l'écran et à la résolution du standard : 640 x 480 pixels. Pour des raisons de lisibilité, nous avons dû limiter le nombre de paramètres de l'espace des observations à 7. Pour la même raison, nous ne pouvons afficher un nombre trop important de points puisque la taille de la fenêtre d'affichage est de 511x357 pixels.

IV.2.4.2 Limitations de taille mémoire

La mémoire vive accessible est de 640 K octets : en effet, le D.O.S. ne peut accéder aux extensions de mémoire que lorsque celles-ci sont déclarées en disque virtuel ou en EMS. Après le lancement du D.O.S. , celle-ci n'est plus que de 553088 octets.

Le driver METAWINDOW™ occupe, quant à lui 105 K octets, et notre programme exécutable 83 K octets.

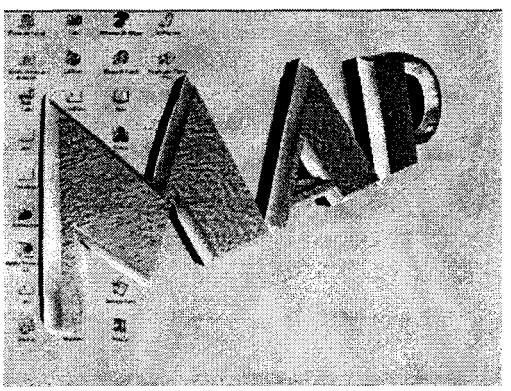
Le programme utilise, quant à lui, environs 4 K octets pour sa gestion interne.

Il nous reste donc en tout environs 350 K octets utilisables pour les données, ce qui est relativement peu, lorsque l'on sait qu'un fichier de 1500 observations en 7 dimensions, exprimé en réels double précision, occupe à lui seul, 82 K octets. Or notre logiciel utilise trois fichiers de ce type.

IV.2.4.3 L'aspect « fermé » de l'application

Notre prototype est écrit pour un environnement DOS amélioré par les routines de Metawindows, par conséquent, aucune ouverture n'est possible vers la majorité des applications actuelles qui fonctionnent sous un environnement graphique. C'est une application « fermée »

IV.3 Implantation *MS Windows* : Le logiciel MAP



Après avoir testé la validité de notre mode de représentation, l'ergonomie de notre interface et effectué quelques essais de classification sur des fichiers synthétiques ainsi que réels (Cf. Chap. VI), nous sommes passés à la phase de l'implantation finale sous Ms Windows. Celle-ci nous a permis de mettre en application la grande majorité des principes d'ergonomie que nous avons passés en revue dans le chapitre IV. Nous avons apporté, avec cette nouvelle application, de nettes améliorations par rapport à l'ébauche sous Metawindows. Nous passons en revue ces points dans le paragraphe suivant.

IV.3.1 *Les améliorations apportées par le développement sous Windows*

. MSWindows permet au développeur de concevoir un logiciel conforme à la majorité des principes d'ergonomie mis en œuvre dans les environnements graphiques modernes, en lui fournissant toute une palette d'objets et de fonctions préexistantes. MSWindows permet ainsi une bonne homogénéité des différentes applications entre elles. Notre application définitive bénéficie de cet environnement standard. Les éléments nouveaux apportés par Windows sont de plusieurs sortes:

☞ **Retour d'information**

- La barre de titre de la fenêtre principale indique le nom du fichier ouvert ainsi que l'espace occupé en mémoire vive et sur disque (Fig. 39-❶).

- La barre de statut indique les processus en cours et leur progression, comme une classification ou un chargement de fichier (Fig. 39-❽)

$$\vec{W} \ / \ \|\vec{W}\| = 1, \text{ et } \vec{W} \perp \vec{V}_{ref} \Leftrightarrow \vec{W} \cdot \vec{V}_{ref} = 0, \qquad \textbf{Equ. 62}$$

et on se place dans le plan engendré par $\vec{V}_{ref}$ et $\vec{W}$.

On utilise comme repère dans ce plan, le repère $(O, \vec{V}_{ref}, \vec{W})$ (Fig. 14).

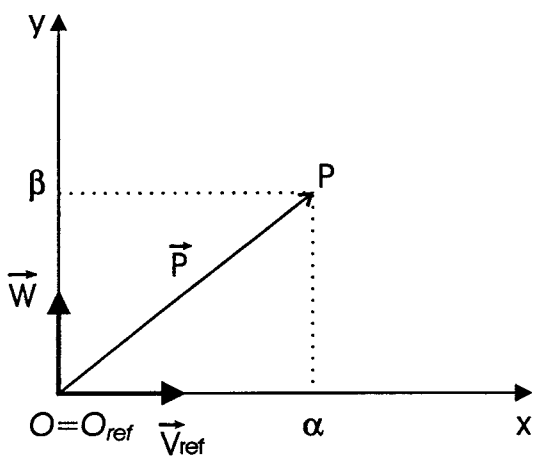


Fig. 15

A un point P de ce plan est attaché le vecteur $\vec{P}$ vérifiant donc:

$$\vec{P} = \alpha . \vec{V}_{ref} + \beta . \vec{W} \qquad \textbf{Equ. 63}$$

Si le point P du plan considéré appartient à l'hypersurface, on a donc :

$$\textbf{arccos} \frac{\left(\vec{V}_{ref} \cdot \vec{P} \right)}{\|\vec{P}\|} = a . \|\vec{P}\| + b \qquad \textbf{Equ. 64}$$

Comme

$$\|\vec{W}\| = \|\vec{V}_{ref}\| = 1, \text{ et } \vec{W} \cdot \vec{V}_{ref} = 0,$$

alors

$$\|\vec{P}\|^2 = \vec{P} \cdot \vec{P} = (\alpha . \vec{V}_{ref} + \beta . \vec{W}) \cdot (\alpha . \vec{V}_{ref} + \beta . \vec{W}) = \alpha^2 + \beta^2 \qquad \textbf{Equ. 65}$$

et :

$$\vec{P} \cdot \vec{V}_{ref} = \alpha . \qquad \textbf{Equ. 66}$$

On obtient ainsi l'équation de la section de l'hypersurface:

$$\boxed{\arccos\frac{\alpha}{\sqrt{\alpha^2+\beta^2}} = a\sqrt{\alpha^2+\beta^2}+b} \qquad \text{Equ. 67}$$

Si, dans le plan $(O,\vec{V}_{ref},\vec{W})$, on utilise le système de coordonnées polaires, le point P est repéré par le couple (ρ,θ)[27], où θ est l'angle $(\vec{V}_{ref},\overrightarrow{OP})$, et ρ la norme du vecteur $\vec{OP}$, c'est à dire $\|\vec{P}\| = \sqrt{\alpha^2+\beta^2}$.

La section est donc définie en coordonnées polaires par une équation affine liant le rayon vecteur $\vec{p}^*$ et l'angle polaire θ.

L'équation de la section en coordonnées polaires est:

$$\begin{cases}\rho = C^{te} \\ 0 \le \theta \le \pi\end{cases} \qquad \text{Equ. 68}$$

pour des droites de décision verticales de la forme $x = C^{te}$,

et

$$\begin{cases}\theta = a\,\rho + b \\ \text{avec } 0 \le \theta \le \pi \quad \text{(Cf. §IV.4.2)} \\ \quad \text{et } \rho \ge 0\end{cases} \qquad \text{Equ. 69}$$

pour les autres. On note que cette courbe, lorsqu'elle n'est pas bornée, est connue sous le nom de « spirale d'Archimède ».

Mis à part les cas où, dans l'espace de représentation, la droite de décision a pour équation $y=C^{te}$ ou $x=C^{te}$, la surface ou l'hypersurface de décision[28] est

---

[27] avec les notations traditionnelles en Mathématiques

donc engendrée par la rotation d'un arc de spirale d'Archimède autour de la droite admettant pour direction le vecteur de référence $\vec{V}_{ref}$ (Cf. §IV.4.3).

Nous présentons ici un aperçu de quelques coupes de sections[29], ainsi que des surfaces tridimensionnelles engendrées, dans des cas types de segments de décisions dans le plan de représentation. Ces courbes ont été obtenues à l'aide du logiciel Matlab® 4.2.

Remarque

Dans toutes ces figures, la surface est engendrée par une rotation de la courbe le long de l'abscisse, admettant $\vec{V}_{ref}$ pour direction (Cf. Fig.15)

III.4.5 *Exemples*

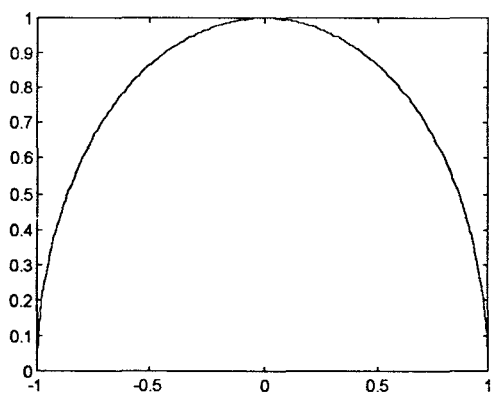
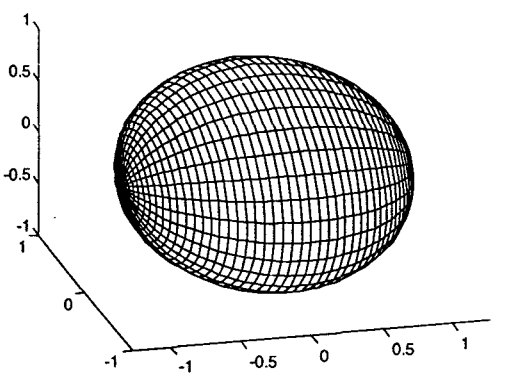


Fig. 16: $\rho = 1$



La surface engendrée est une sphère

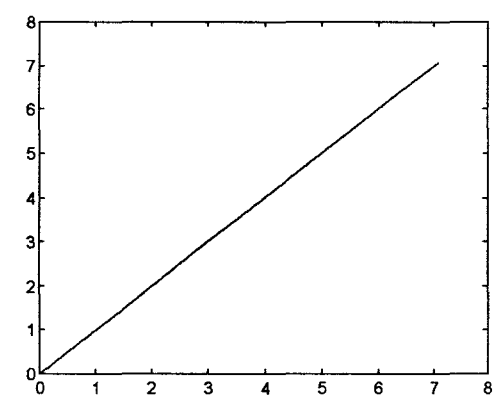
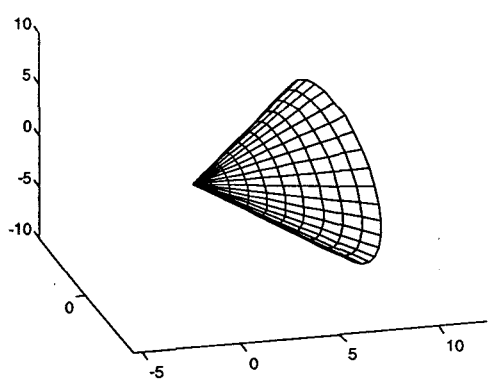


Fig. 17: $\alpha = \pi/4$



La surface engendrée est un cône

---

[29] sections de l'hypersurface de décision sections par des plans engendrés par deux vecteurs dont $\vec{V}_{ref}$ et passant par O_{ref}.

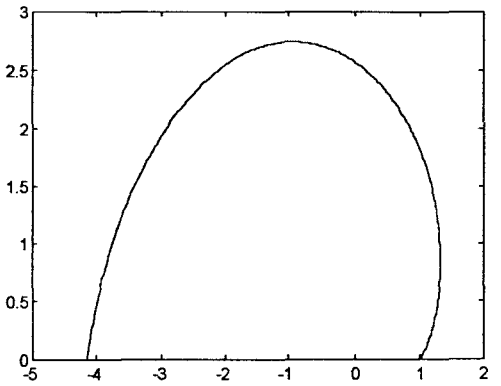
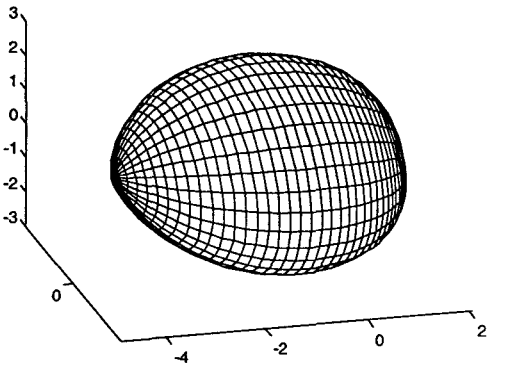


Fig. 18: $\alpha = \rho - 1$



La surface engendrée est en forme de toupie

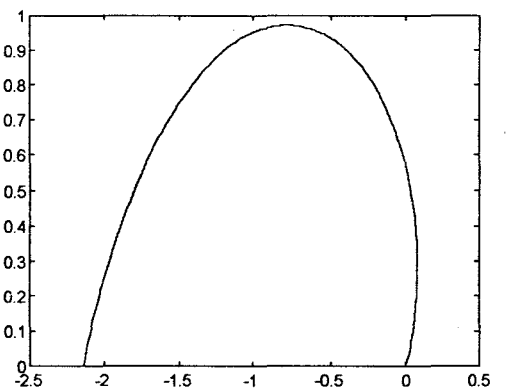
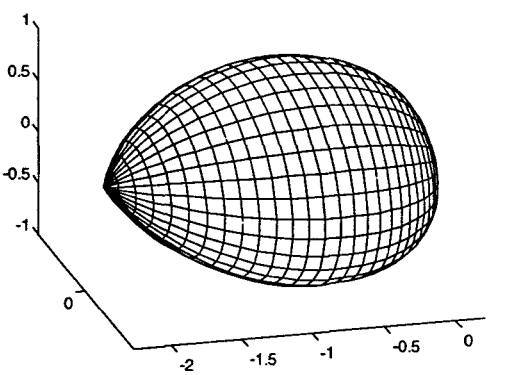


Fig. 19: $\alpha = \rho + 1$



Surface engendrée par Fig.19

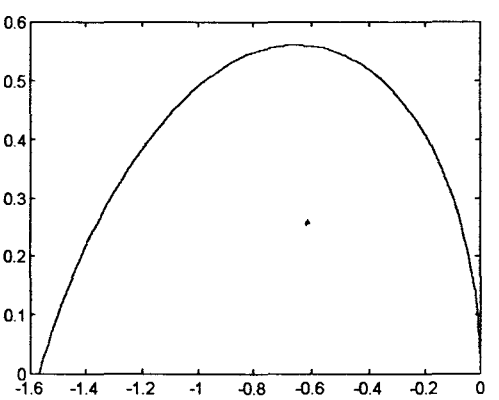
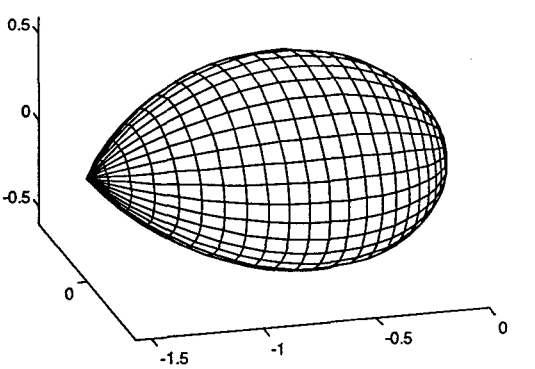


Fig. 20: $\alpha = \rho + \pi/2$



Surface engendrée par Fig. 20

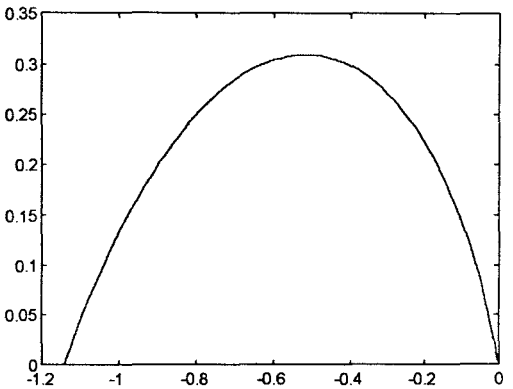
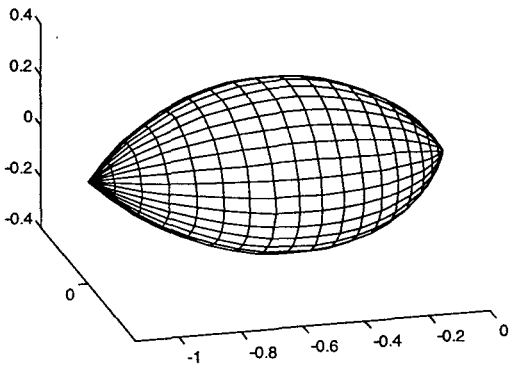


Fig. 21: $\alpha = \rho + 2$



Surface engendrée par Fig. 21

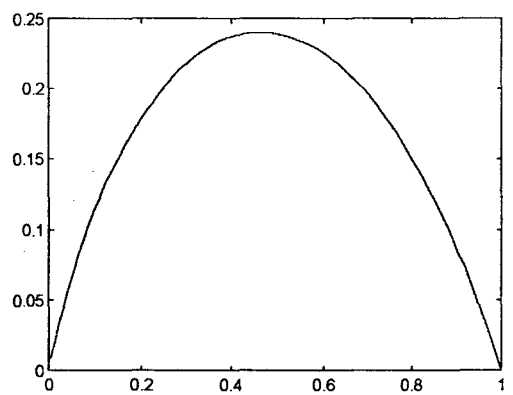
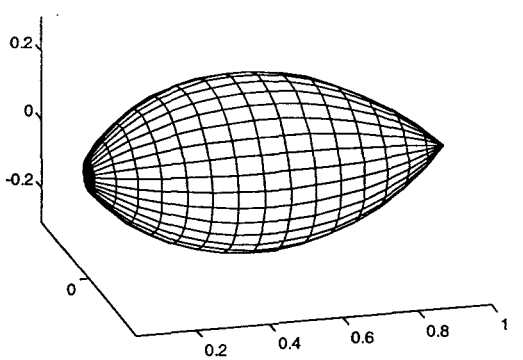


Fig. 22: $\alpha = -\rho + 1$



Surface engendrée par Fig. 22

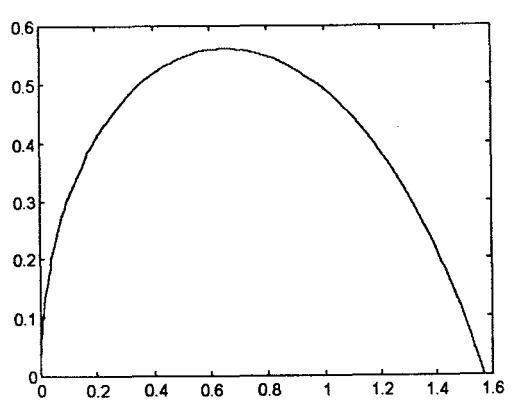
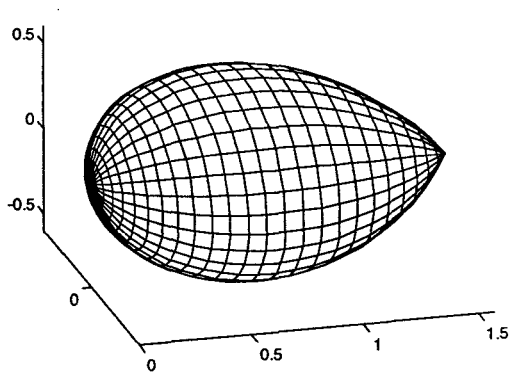


Fig. 23: $\alpha = -\rho + \pi/2$



Surface engendrée par Fig. 23

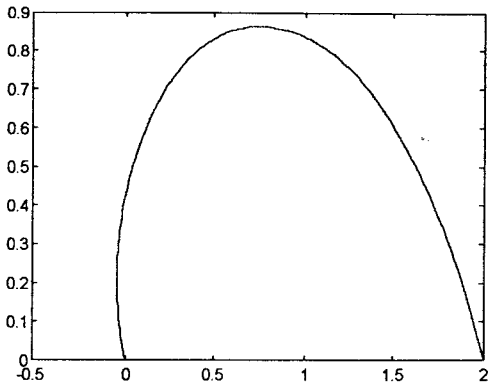
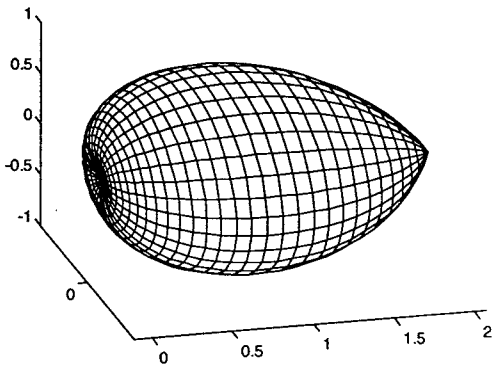


Fig. 24: $\alpha = -\rho + 2$



Surface engendrée par Fig. 24

III.4.6 *Conclusion*

Dans ce chapitre, nous avons passé en revue les aspects les plus significatifs de notre algorithme de représentation. Il s'avère, après cette étude, que cet algorithme est valide, et qu'il présente des potentialités très intéressantes. Mais, contrairement à de multiples algorithmes de classification, une simple implantation sur calculateur ne permettrait pas d'exploiter de manière exhaustive ses capacités. En effet, lorsque nous avons envisagé cet algorithme, c'était dans le but qu'un utilisateur lambda, ne possédant pas forcément de connaissances en analyse de données et ne disposant pas d'un système particulièrement performant et onéreux, puisse aborder la classification d'un ensemble d'observations de manière simple, rapide et intuitive, par le biais de notre algorithme.

Pour cela, il est primordial de concevoir une interface utilisateur aussi agréable et simple que possible à utiliser.

C'est ce sujet que nous traitons au chapitre suivant où nous abordons quelques points importants d'ergonomie en matière de conception de logiciels.

CHAPITRE IV
LES IMPLANTATIONS DE LA REPRESENTATION ANGULAIRE

IV. LES IMPLANTATIONS DE LA REPRESENTATION ANGULAIRE

IV.1 Généralités

☞ Choix du système

Il est aisé d'expliquer le choix du PC pour le développement de notre application : en effet, l'importance de l'implantation de ce dernier dans tous les secteurs de notre société est bien connue, et nous voulions créer une application qui soit accessible au plus grand nombre, et non plus réservée aux utilisateurs universitaire de gros systèmes, comme c'est souvent le cas en traitement de données.
Pour exploiter pleinement le type de représentation plane que nous avons choisi, l'interface graphique s'impose[JMG95] : il faut permettre à l'opérateur de manipuler le référentiel mobile (à savoir le point d'origine et la direction du vecteur de référence) de la manière la plus simple et naturelle possible. L'utilisateur doit pouvoir se concentrer sur les données qu'il étudie sans avoir à se préoccuper du logiciel qu'il utilise : c'est le principe d'ergonomie de transparence bien connue des concepteurs d'interfaces graphiques.

Nous nous sommes donc attachés à ce que le logiciel conçu soit le plus *transparent* et ergonomique possible, par un choix adéquat de la manière dont les informations sont affichées et les commandes saisies.

☞ Choix du langage

Pour implanter notre concept, nous avons choisi le langage C, et ceci pour deux raisons :
- Le langage C est connu pour sa rapidité d'exécution, et la compacité des programmes créés à l'aide de celui-ci sur de petits systèmes. C'est un aspect primordial, étant donnée la masse très souvent importante des données à analyser.
- Le langage C est proche du langage machine sous bien des aspects, et il présente une grande puissance pour le traitement des pointeurs, ce qui, nous le verrons plus tard, nous intéresse tout particulièrement pour la gestion de bases de données importantes. De plus, la plupart des logiciels actuels sont écrits en C ou en C++ (orienté objet)

☞ Choix du support graphique

☞ Un prototype a été implanté à l'aide du logiciel utilitaire de programmation **METAWINDOW™**.

* **METAWINDOW™** fournit une base de procédures permettant à la fois le multi-fenêtrage, l'utilisation de la souris, et les tests d'inclusion de points dans des figures pré-définies, dans un programme tournant sous DOS. Cet environnement simple suffisait donc amplement pour une première étude de notre principe de représentation angulaire

☞ Le logiciel définitif a été conçu en langage C sous Windows.

* **MS Windows** fournit en effet un environnement graphique parfaitement adapté à l'implantation et aux extensions ultérieures de notre application. L'environnement Windows permet, en application des principes de base d'ergonomie, une utilisation transparente et intuitive des logiciels, tout en fournissant de surplus une palette de fonctionnalités et d'objets préexistants permettant l'écriture d'un logiciel parfaitement ergonomique.

☞ Cependant, nous avions tout d'abord besoin de pouvoir mettre en oeuvre la plupart des idées de bases du logiciel projeté, sans la lourdeur d'apprentissage qu'impliquait une implantation sous MS Windows afin de tester la validité de nos idées sur un prototype plus simple.

IV.2 Implantation sous Metawindows : le programme INTERACT

IV.2.*1* *Le retour d'information*

IV.2.*1*.1 Mode d'affichage des coordonnées des point et vecteur de référence

Afin de fournir à l'utilisateur un retour d'information « visuel » sur l'état du référentiel, nous avions d'abord pensé visualiser les positions respectives de l'origine du référentiel et du vecteur de référence à l'aide de diagrammes de type « toiles d'araignées » (Cf. Fig. 25).

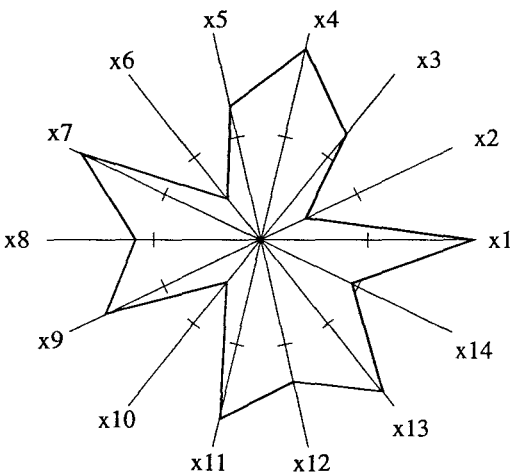


Fig. 25 : Exemple de toile d'araignée de dimension 14

Ce type de diagramme, souvent utilisé lorsque la dimension de l'espace des paramètres dépasse trois, présente l'avantage d'être très *parlant* pour l'utilisateur. Ce dernier peut aisément retenir visuellement une configuration des paramètres définissant le référentiel mobile dans l'espace multidimensionnel.

Cependant, l'aspect angulaire de la position du vecteur référence s'est révélé mal traduit. De plus, étant donnée la taille de l'écran ainsi que l'emplacement que l'on devait réserver à la représentation des observations, ce type d'affichage serait vite devenu illisible pour un nombre important de paramètres. C'est pourquoi nous avons finalement opté pour un type d'affichage plus simple et plus facile à manier :

- La position de l'origine est représentée par des curseurs se déplaçant sur des *barrettes* (Cf. Fig. 26-❸ et 27).

- Pour représenter de manière la plus imagée possible la position du vecteur de référence, en coordonnées angulaires généralisées, il fallait opter pour un affichage faisant intervenir un angle. C'est pourquoi cette position est affichée à l'aide de

portions de disques ou *camemberts* (Cf. Fig. 26-❷ et 27). Si la dimension de l'espace des observations est **n**, les *camemberts* seront au nombre de **n-1** (Cf. §III.2.2)

Les barrettes et les portions de disques sont de couleurs différentes pour chaque axe de référence, permettant ainsi une meilleure lisibilité pour l'opérateur.

L'interface d'INTERACT se présente comme schématisée sur la Figure 26.

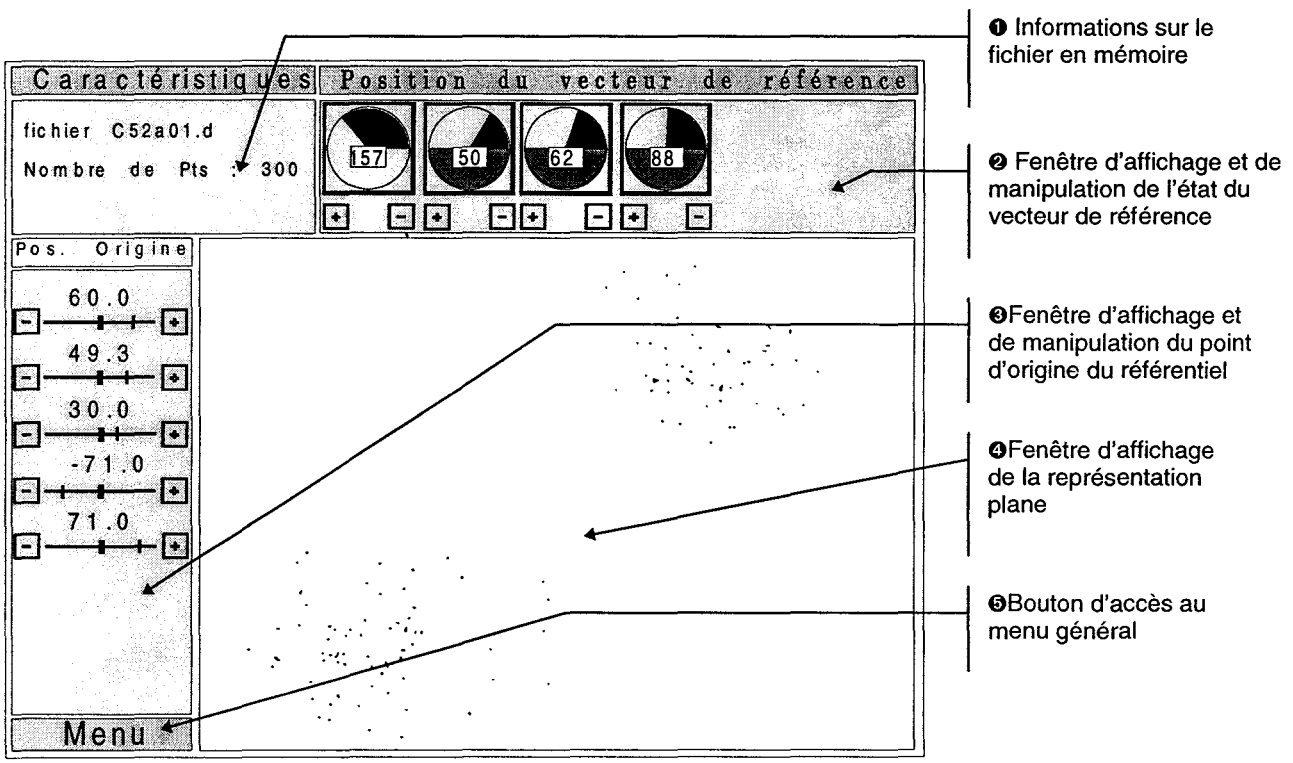


Fig. 26 : Interface Graphique de l'application sous Metawindows

Ces modes d'affichage des positions respectives de l'origine et du vecteur de référence possèdent, par rapport aux diagrammes évoqués précédemment, l'avantage de pouvoir être utilisés pour des espaces de dimension importante, tout en gardant leur lisibilité. Cependant, bien que ces deux modes d'affichage permettent de bien percevoir les modifications opérées sur le référentiel, ils ne donnent pas de renseignements assez précis sur les positions respectives de l'origine de ce dernier, ni du vecteur de référence. C'est pourquoi, toujours dans le souci du retour d'information vers l'utilisateur, nous avons aussi affiché au niveau de chaque *barrette* et de chaque *camembert*, les valeurs numériques de la variable correspondante.

IV.2.1.2 Affichage des valeurs numériques

♦ Origine

La latitude L_i de déplacement de l'origine, sur l'axe i, est la fourchette des valeurs que peut prendre la coordonnée d'ordre i de l'origine de notre référentiel. Celle-ci est fixée de la manière suivante:

$$L_i = \max(|x_i \max|, |x_i \min|)$$

où x_i **max**, et x_i **min** sont respectivement les valeurs maximale et minimale signées du $i^{ème}$ paramètre (x_i min peut être négatif et supérieur en valeur absolue à x_i max).

Pour afficher la valeur numérique du décalage de l'origine, nous avons cru bon d'effectuer une normalisation (à l'affichage uniquement), afin que l'utilisateur puisse effectuer des comparaisons entre les différents paramètres.

Cette normalisation est effectuée en multipliant la valeur du décalage par un facteur d'échelle E_i avec : $E_i = 100 / L_i$ (Cf. Fig. 28)

♦ Vecteur de référence

La valeur, en degrés, de chaque coordonnée angulaire du vecteur de référence est affichée en surimpression sur le *camembert* correspondant. (Cf. Fig. 27)

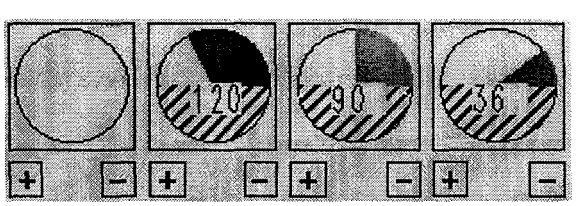


Fig. 27: Vecteur de référence

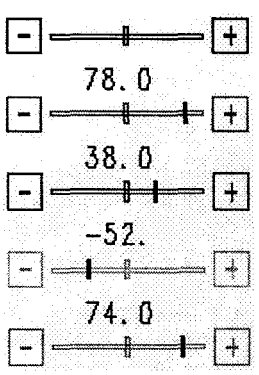


Fig. 28: Origine

IV.2.1.3 Visualisation des points de la représentation plane

Après l'évaluation, par une procédure de comparaison, des extrema en abscisse et en ordonnée, de l'ensemble des points images des observations, l'affichage est cadré à partir de ces extrema, de manière à ce que l'ensemble des pixels images obtenus emplisse la fenêtre de représentation (Cf. Fig.29) ; i.e.:

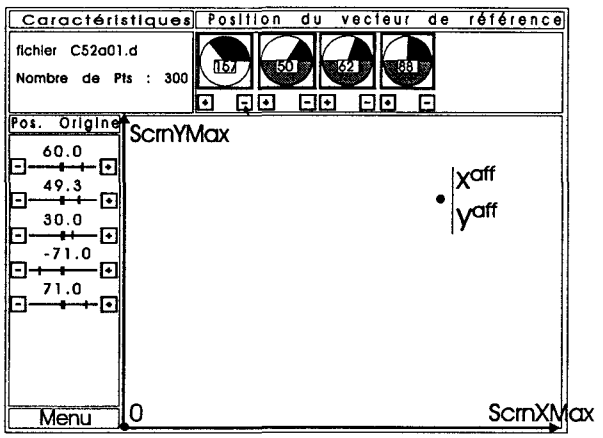


Fig. 29 : Zone Ecran de visualisation des points images

- $x^{aff} = (x - xmin) . \frac{ScrnXMax}{xMax - xmin}$
-
- $y^{aff} = (y - ymin) . \frac{ScrnYMax}{yMax - ymin}$

avec :

- ScrnXMax, ScrnYMax : maxima des coordonnées de l'écran (les minima étant nuls),
- x^{aff}, y^{aff} : coordonnées du pixel image (pixel écran correspondant au point image d'une observation par la transformation angulaire)
- x , y : coordonnées du point image d'une observation par la transformation angulaire.
- xmin, xMax, ymin, yMax : extrema des points images des observations.

par la transformation angulaire.

IV.2.1.4 Affichage des paramètres d'une observation

Lors des premiers essais du logiciel, nous déplacions l'origine du référentiel en jugeant de la qualité du décalage uniquement sur l'apparence de la représentation des données. Ce type de modification du référentiel s'est vite avéré insuffisant. C'est pourquoi nous avons ajouté une aide au déplacement de l'origine, en permettant au praticien d'afficher les coordonnées d'un point désigné avec la souris. Cette possibilité, qui permet à l'analyste de connaître les coordonnées de l'observation

multidimensionnelle associée au pixel image sélectionné, résulte de l'exploitation de la structure particulière des fichiers.

L'opérateur indique l'emplacement du pixel image choisi avec le bouton gauche de la souris, puis valide sa demande avec le bouton droit. Les coordonnées normalisées de l'observation multidimensionnelle associée (Cf. §V.2.1.2) sont alors affichées à l'emplacement qu'occupe le curseur lors de la validation (Cf. Fig. 30).

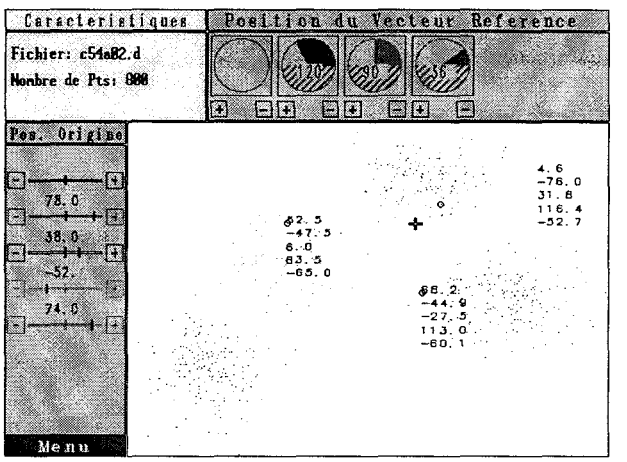


Fig. 30: exemple d'affichage des paramètres de trois observations

IV.2.2 *Les fonctions de l'application INTERACT*

Le prototype INTERACT sert à valider notre concept de représentation plane afin de prouver sa faisabilité.

Les fonctions qui ont été implantées sont donc des fonctions permettant l'affichage de la représentation Module/Angle, ainsi que les fonctions de base permettant la classification à partir d'une ou de plusieurs représentations d'un même ensemble d'observations[ESS93].

IV.2.2.1 Saisie des fichiers d'observations

Cette manoeuvre est réduite au strict minimum dans le cas de notre prototype. La saisie d'un fichier initial se fait automatiquement au lancement du programme. Pour effectuer la saisie d'un nouveau fichier en cours de programme, il suffit de sélectionner le menu général (Fig. 26-❺, Fig. 31), avec la souris, et de la même manière, de sélectionner dans ce menu la saisie d'un nouveau fichier.

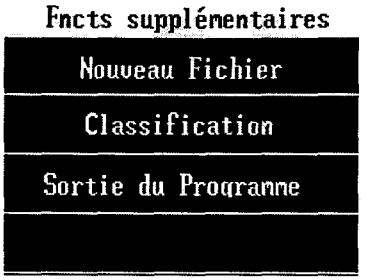


Fig. 31: Menu général

IV.2.2.2 Modification de l'état du référentiel / Affichage des points.

L'opérateur modifie la position du vecteur de référence, ou de l'origine, en *cliquant* avec la souris dans les carrés d'incrémentation ou de décrémentation qui sont placés au niveau de l'affichage des grandeurs correspondantes (Cf. Fig. 26-❷, 26-❸, 27 et 28).

On peut déplacer l'origine de $-L_i$ à $+L_i$ sur l'axe I_i (Cf. §V.2.1.2), et les coordonnées angulaires du vecteur de référence sont toutes comprises entre 0° et 180°, sauf le premier angle qui est compris entre 0° et 360°, ceci, à l'instar des coordonnées sphériques, pour éviter les redondances.

Il y a deux modes de modification du référentiel :

- En mode *pas à pas*, avec le bouton de gauche de la souris, à chaque modification du référentiel, l'indicateur de position correspondant est remis à jour, la représentation Module/Angle est recalculée, et l'affichage rafraîchi.
- En mode *différé* avec le bouton de droite de la souris, tant que le bouton est enfoncé, nous incrémentons ou décrémentons la grandeur sélectionnée (avec un pas plus petit qu'en vitesse lente). Seul l'indicateur de position correspondant évolue. Les calculs, ainsi que le rafraîchissement de la représentation plane, ne sont effectués qu'après relâchement du bouton droit de la souris. (Cette formule s'est avéré nécessaire à cause de l'augmentation du temps de calcul de la représentation pour des bases de données importantes)

IV.2.2.3 Fonctions de classification interactive

Le logiciel comprend quatre fonctions de gestion de classes qui sont activées par le choix du bloc de classification dans le menu général. On active les procédures de classification en *cliquant* avec la souris dans la case correspondante du menu

général(Fig. 32). Les fonctions de modification du référentiel mobile sont alors invalidées et la fenêtre « routines de classification » remplace alors la fenêtre de manipulation du vecteur de référence (Cf. Fig. 32

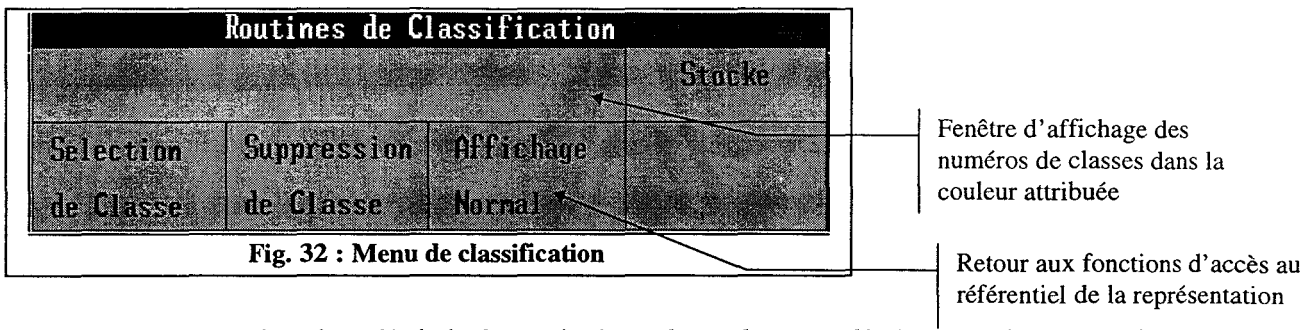


Fig. 32 : Menu de classification

☞ Pour des raisons de simplicité, la création des classes d'observations se fait en entourant les pixels images sélectionnés avec un polygone comprenant 20 sommets au maximum, créé à l'aide de la souris : l'utilisateur choisit l'emplacement du sommet à l'aide du bouton de gauche, puis valide l'emplacement choisi avec le bouton de droite. Enfin, il indique le dernier sommet, en pressant simultanément sur les deux boutons de la souris (Cf. Fig.33).

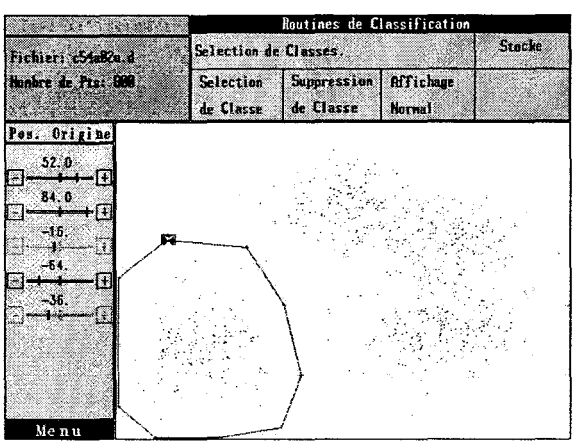


Fig. 33 : Création d'une classe d'observations

Un test d'inclusion des pixels images dans la figure ainsi créée est alors effectué. Chaque fois qu'un pixel image situé à l'intérieur du polygone est détecté, l'observation dont il est la représentation est assignée à la classe en cours de création, et déplacée dans une zone de mémoire associée à cette classe, et ainsi de suite jusqu'au dernier pixel image de l'ensemble des observations.

La procédure de réorganisation des classes attribue alors aux points sélectionnés une couleur spécifique, et les réaffiche dans cette couleur. Le numéro de la classe, et le nombre de points qu'elle contient sont affichés

Chaque fois qu'une nouvelle classe est créée, la base de données en mémoire vive est restructurée. Si par exemple, l'opérateur a créé deux classes d'observations, la base de données sera structurée en trois ensembles : les deux classes créées seront en tête, et la dernière « classe », sera en fait, l'ensemble des points non classés.

☞ La procédure de suppression de classe permet à l'utilisateur de refondre une classe d'observations qu'il a créée avec l'ensemble des observations non classées (Cf. Fig.33). Cette procédure ne peut bien sûr être lancée que si le nombre de classes est supérieur à 1, la dernière *classe* étant, par convention, le groupe des observations non classées, et ne pouvant être supprimée.

Lorsque la suppression de classe est activée, la procédure affiche le numéro de chaque classe existante, dans la couleur qui lui a été attribuée. Il suffit d'entrer au clavier le numéro de la classe à supprimer. Les observations appartenant à cette classe, sont alors assignées à la dernière *classe* des points non classés, et les pixels images correspondants sont réaffichés dans la couleur du groupe des observations non classées.

☞ Lorsque l'on est satisfait d'une classification ainsi effectuée, on peut stocker les observations reclassées, dans un fichier en mémoire de masse. Ce fichier est indépendant du fichier original des observations, et porte le même nom que ce dernier, avec une extension différente : l'extension " .cla".

☞ Sécurités

Le mode de classification choisi interdit de placer dans une nouvelle classe un point déjà classé. Lorsque l'on veut stocker en mémoire de masse une nouvelle classification, et qu'un classement y existait déjà, une fenêtre est affichée, demandant de confirmer la requête. Si celle-ci est confirmée, le fichier existant est écrasé par le nouveau fichier créé.

IV.2.3 *Quelques détails de Programmation*

IV.2.3.1 Calcul des coordonnées cartésiennes du vecteur de référence

On rappelle que la position du vecteur de référence est fixée par le biais de ses coordonnées angulaires généralisées, mais que tous les calculs sont effectués en coordonnées cartésiennes, d'où la nécessité d'une conversion.

On a vu que si $v_1,\ldots, v_n$ sont les coordonnées cartésiennes du vecteur de référence $\vec{V}_{ref}$, et ς, $\theta_2, \ldots, \theta_n$ ses coordonnées angulaires généralisées,

$v_1,\ldots, v_n$ peuvent s'exprimer comme suit :

$$v_n = \varsigma.\cos\theta_n$$
$$v_{n-1} = \varsigma.\sin\theta_n \,.\, \cos\theta_{n-1}$$
$$v_{n-2} = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \cos\theta_{n-2}$$
$$\text{"} \quad \text{"}$$
$$\text{"} \quad \text{"}$$
$$v_2 = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,.\,.\, \cos\theta_2$$
$$v_1 = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,.\,.\, \sin\theta_2$$

- $\vec{I}_1,\vec{I}_2,\vec{I}_3,\ldots,\vec{I}_n$ sont les vecteurs de la base de l'espace cartésien des observations.

- θ_{n-i} est l'angle que fait la projection du vecteur $\vec{V}_{ref}$ dans l'espace de dimension n-i défini, par les axes $\vec{I}_1,\vec{I}_2,\vec{I}_3,\ldots,\vec{I}_{n-i}$, avec l'axe $\vec{I}_{n-i}$. (Cf. §III.2.2)

On remarque ici l'aspect répétitif de ces formules. Nous nous sommes donc appliqués à trouver une programmation récurrente pour le calcul des coordonnées cartésiennes du vecteur de référence $\vec{V}_{ref}$ à partir de ses coordonnées angulaires, sa norme étant, par principe, fixée à 1.

☞ On peut construire les coordonnées cartésiennes du vecteur de référence, à partir des coordonnées angulaires généralisées, en incrémentant la dimension de l'espace par étapes. Le calcul sera effectué comme suit :

1ère étape : Coordonnées polaires dans le sous espace de base I_1, I_2

$$v_2 = \varsigma \,.\, \cos\theta_2$$
$$v_1 = \varsigma \,.\, \sin\theta_2$$

2ième étape : Coordonnées sphériques dans le sous espace de base I_1,I_2, I_3

$$v_3 = \varsigma \,.\, \cos\theta_3$$
$$v_2 = \varsigma \,.\, \sin\theta_3 \,.\, \cos\theta_2$$
$$v_1 = \varsigma \,.\, \sin\theta_3 \,.\, \sin\theta_2$$

.
.

jième étape : Dimension j+1, sous espace de base I_1, I_2,... , I_{j+1}

$$v_{j+1} = \varsigma.\cos\theta_{j+1}$$
$$v_j = \varsigma.\sin\theta_{j+1} \,.\, \cos\theta_j$$
" "
" "
$$v_2 = \varsigma.\sin\theta_{j+1} \,.\, \sin\theta_j \,.\, \sin\theta_{j-1} \,./.../\, .\, \cos\theta_2$$
$$v_1 = \varsigma.\sin\theta_{j+1} \,.\, \sin\theta_j \,.\, \sin\theta_{j-1} \,./.../\, .\, \sin\theta_2$$
.
.

Etape finale : Dimension ***n***, espace d'origine, de base I_1, I_2, ... , I_n

$$v_n = \varsigma.\cos\theta_n$$
$$v_{n-1} = \varsigma.\sin\theta_n \,.\, \cos\theta_{n-1}$$
$$v_{n-2} = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \cos\theta_{n-2}$$
" "
" "
$$v_2 = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,./.../\, .\, \cos\theta_2$$
$$v_1 = \varsigma.\sin\theta_n \,.\, \sin\theta_{n-1} \,.\, \sin\theta_{n-2} \,./.../\, .\, \sin\theta_2$$

L'algorithme de calcul correspondant sera donc le suivant (Fig. 35):

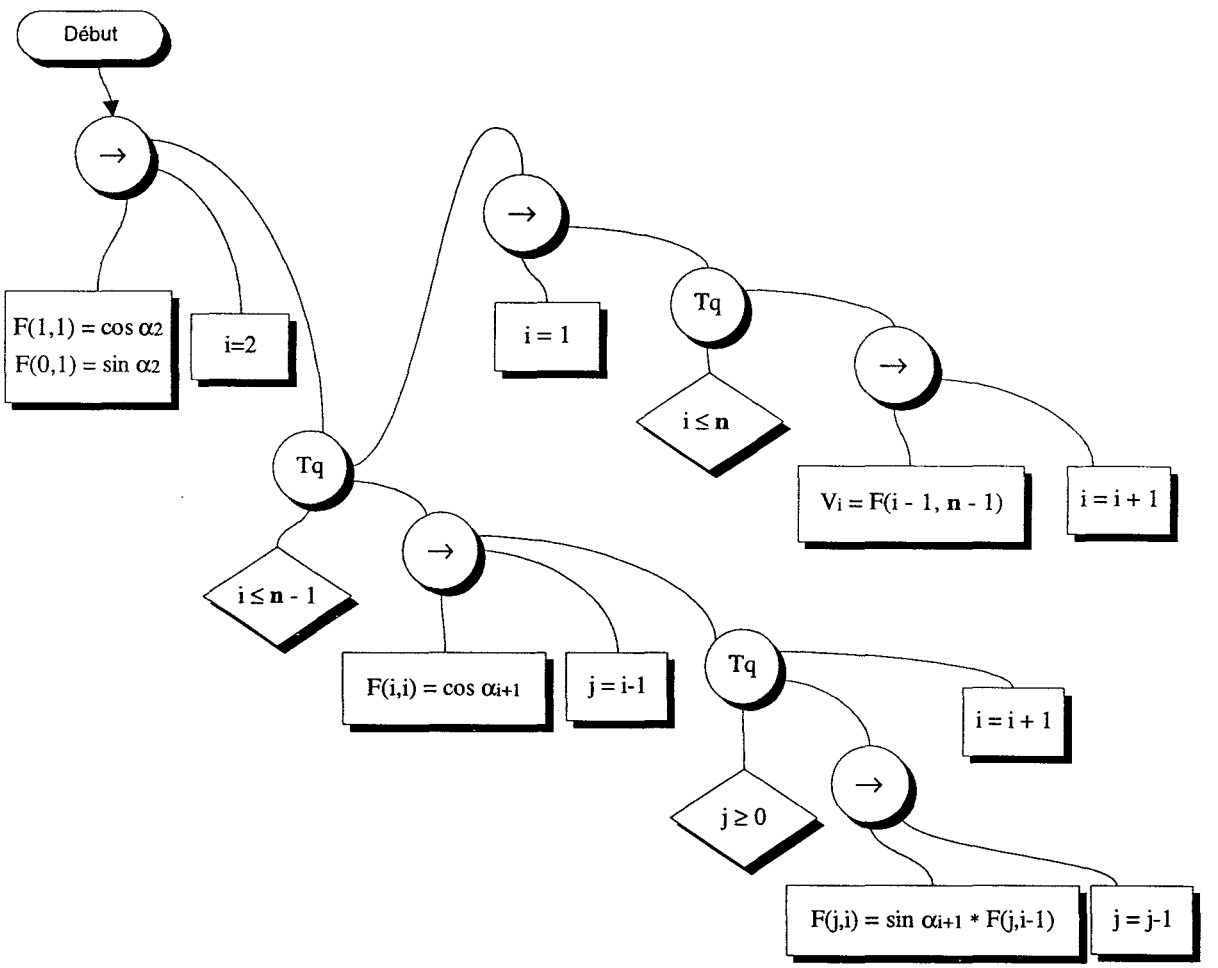


Fig. 34 Arbre programmatique de calcul des coordonnées du vecteur de référence

IV.2.3.2 Organisation des données

IV.2.3.2.a)*Structure des fichiers de données en mémoire de masse*

La structure de base des fichiers de données, ainsi que sa forme, reprend celle utilisée actuellement dans l'équipe de chercheurs « image et décision » du Centre d'Automatique de Lille. Cette structure a l'avantage d'être transportable aisément, puisque les données sont stockées sous la forme de caractères ASCII. Elle permet par ailleurs une grande souplesse d'utilisation, puisque les bases de données classées et non classées sont structurées de la même manière.

Les données apparaissent comme suit:

$\mathbf{p}, \mathbf{n}, \mathbf{K}$

$\mathbf{p}_1, \cdots, \mathbf{p}_K$

$x_1(1), \ldots, x_n(1)$
.
.
$x_1(\mathbf{p}_1), \ldots, x_n(\mathbf{p}_1)$
.
.
$x_1(\mathbf{p}_1+\mathbf{p}_2), \ldots, x_n(\mathbf{p}_1+\mathbf{p}_2)$
.
.
.
$x_1(\mathbf{p}), \ldots, x_n(\mathbf{p})$

Avec **p** : nombre total d'observations, ***n*** : dimension de l'espace des paramètres, **K** : nombre de classes (dans le cas d'un fichier d'observations non classées, **K** = 1), $\mathbf{p}_1, \ldots, \mathbf{p}_K$: nombre d'observations par classe, $x_j(i)$: paramètre d'ordre j de la $i^{ième}$ observation.

☞ **Structure des données en mémoire vive**

En mémoire vive, les données sont aussi stockées sous une forme structurée, à l'aide de pointeurs. Les observations et leurs représentations planes sous forme de points, ont le même indice, ce qui permet de retrouver aisément l'observation originale à partir de sa représentation.

Il n'est pas rare d'avoir à traiter des bases de données de plusieurs centaines observations, dans un espace de plus de 3 dimensions. Les observations sont stockées en mémoire dans des variables de type réel double précision, donc pour un tableau de 1500 observations en 7 dimensions, l'occupation mémoire sera de 82 K octets. Or le langage C sous système DOS ne peut pas gérer, avec 8 bits d'adressage, des tableaux de plus de 64 K octets chacun.

C'est pourquoi nous avons choisi de stocker ces données en mémoire vive, sous forme de blocs structurés adressés par pointeurs. Nous aurons 4 blocs organisés de la sorte :

- Le bloc des observations originales,
- Le bloc des observations, après le changement d'origine de l'espace multidimensionnel (Cf. Chap. III),
- Le bloc de la transformée plane des observations,
- Le bloc des pixels image résultants.

Les données des deux premiers blocs sont ordonnées de la même manière que dans le fichier original des observations en mémoire de masse.

Ce mode de stockage a présenté un avantage très net de souplesse par rapport au stockage dans des tableaux de format fixe, lors de la conception des procédures de classification interactive. En effet :

- Bien qu'intrinsèquement, le C ne fasse pas de différence réelle entre tableaux et pointeurs, la gestion directe par pointeurs s'est révélée plus efficace et rapide, avec le compilateur C dont nous disposions, que celle par tableaux. Il en a résulté un gain de temps d'un facteur 10 environs dans le reclassement des points lors de la construction des classes, ainsi que dans le rafraîchissement de la représentation.

- Lors de la suppression d'une classe, ce mode de mémorisation nous a permis de déplacer les données par blocs (opérations de bas niveau mémoire), permettant d'effectuer l'opération de déplacement de la classe supprimée dans le groupe des points non classés, ainsi que le décalage des autres classes, en quelques dixièmes de secondes, alors que la même opération effectuée point par point prenait de une à deux minutes.

IV.2.3.3 Extensions

Nous avons complété le fichier des observations décrit précédemment, par un autre fichier qui porte le même nom, mais auquel nous avons affecté une extension différente : Un fichier « .cla », généré par les fonctions de classification, contient l'ensemble des observations après classification. Ce fichier a, bien entendu, la même structure que le fichier original.

IV.2.4 *Limitations d'INTERACT*

Les limitations de notre prototype sont de trois ordres :

IV.2.4.1 Limitations d'affichage

- Limitations dues au format d'affichage du standard VGA 16 couleurs géré par METAWINDOW™ qui ne comporte que **16** couleurs affichables dans une palette de 64 couleurs (les couleurs sont définies sur un octet). Le nombre effectif de couleurs disponibles est donc de 16, auquel il faut retrancher les couleurs réservées pour les indicateurs de position du référentiel. Nous sommes donc limités à 10 classes d'observations maximum.

- Limitations dues à la taille de l'écran et à la résolution du standard : 640 x 480 pixels. Pour des raisons de lisibilité, nous avons dû limiter le nombre de paramètres de l'espace des observations à 7. Pour la même raison, nous ne pouvons afficher un nombre trop important de points puisque la taille de la fenêtre d'affichage est de 511x357 pixels.

IV.2.4.2 Limitations de taille mémoire

La mémoire vive accessible est de 640 K octets : en effet, le D.O.S. ne peut accéder aux extensions de mémoire que lorsque celles-ci sont déclarées en disque virtuel ou en EMS. Après le lancement du D.O.S. , celle-ci n'est plus que de 553088 octets.

Le driver METAWINDOW™ occupe, quant à lui 105 K octets, et notre programme exécutable 83 K octets.

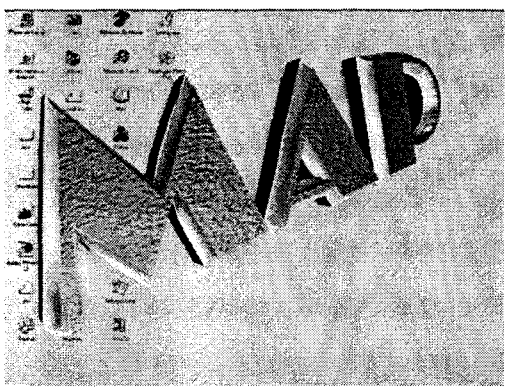
Le programme utilise, quant à lui, environs 4 K octets pour sa gestion interne.

Il nous reste donc en tout environs 350 K octets utilisables pour les données, ce qui est relativement peu, lorsque l'on sait qu'un fichier de 1500 observations en 7 dimensions, exprimé en réels double précision, occupe à lui seul, 82 K octets. Or notre logiciel utilise trois fichiers de ce type.

IV.2.4.3 L'aspect « fermé » de l'application

Notre prototype est écrit pour un environnement DOS amélioré par les routines de Metawindows, par conséquent, aucune ouverture n'est possible vers la majorité des applications actuelles qui fonctionnent sous un environnement graphique. C'est une application « fermée »

IV.3 Implantation *MS Windows* : Le logiciel MAP



Après avoir testé la validité de notre mode de représentation, l'ergonomie de notre interface et effectué quelques essais de classification sur des fichiers synthétiques ainsi que réels (Cf. Chap. VI), nous sommes passés à la phase de l'implantation finale sous Ms Windows. Celle-ci nous a permis de mettre en application la grande majorité des principes d'ergonomie que nous avons passés en revue dans le chapitre IV. Nous avons apporté, avec cette nouvelle application, de nettes améliorations par rapport à l'ébauche sous Metawindows. Nous passons en revue ces points dans le paragraphe suivant.

IV.3.1 *Les améliorations apportées par le développement sous Windows*

. MSWindows permet au développeur de concevoir un logiciel conforme à la majorité des principes d'ergonomie mis en œuvre dans les environnements graphiques modernes, en lui fournissant toute une palette d'objets et de fonctions préexistantes. MSWindows permet ainsi une bonne homogénéité des différentes applications entre elles. Notre application définitive bénéficie de cet environnement standard. Les éléments nouveaux apportés par Windows sont de plusieurs sortes:

☞ **Retour d'information**

- La barre de titre de la fenêtre principale indique le nom du fichier ouvert ainsi que l'espace occupé en mémoire vive et sur disque (Fig. 39-❶).

- La barre de statut indique les processus en cours et leur progression, comme une classification ou un chargement de fichier (Fig. 39-❸)

☞ **Facilité d'utilisation / ergonomie / interactivité**

- La barre de menu permet l'accès rapide aux multiples nouvelles fonctionnalités, organisées par catégorie (Fig. 39-❶)

- L'homogénéité de notre interface avec les autres applications Windows la rend plus facilement et intuitivement manipulable.

- Nous verrons par la suite que le programme est rendu plus interactif par l'usage de boites de dialogue non modales qui permettent une manipulation plus « naturelle » des données.

☞ **Gestion de la mémoire**

- Windows (et Windows95 en particulier) permet une meilleure gestion de l'espace mémoire accessible que le simple DOS. Notre nouvelle application dispose donc de toute la mémoire disponible dans le système. Nous avons pu classer des fichiers de 16000 observations en dimension 3, alors que le prototype ne pouvait traiter que 1500 observations en dimension 5 maximum.

☞ **Ouverture**

Cette ouverture est de deux sortes :

- Ce nouvel environnement de programmation nous permet l'échange de données et d'images avec d'autres applications. De plus, le type même de la programmation, orientée objet et modulaire, permet aisément d'ajouter de nouvelles fonctionnalités à notre application.

- L'ouverture est aussi dirigée vers les utilisateurs potentiels. En effet, l'environnement graphique Windows est très largement majoritaire, à la fois dans le monde de l'entreprise et celui de la recherche. Développer sous Windows permet donc l'accès du plus grand nombre à un logiciel simple et intuitif de classification, notre objectif premier.

IV.3.2 - *Organisation des données MAP vs INTERACT*

Avec MAP, nous avons totalement revu la manière dont les diverses données étaient gérées, aussi bien sur la plan formel que sur le plan de la programmation.

IV.3.2.1 Gestion de la mémoire

☞ Prototype INTERACT

- Avec INTERACT, Les données sont organisées en blocs adressés par des pointeurs, ainsi qu'en tableaux. La plupart de nos variables sont globales, ce qui rend le programme difficilement améliorable ou même extensible. Le prototype gère en mémoire vive quatre blocs principaux permanents de données : le bloc des observations originales, le bloc des observations après changement d'origine du repère, le bloc des représentations planes, et enfin, le bloc des points écrans.

☞ Logiciel MAP

- MAP gère aussi les données par blocs mémoire mais, par contre, l'organisation des divers ensembles de données diffère totalement entre le prototype et le produit finalisé. Il n'existe pratiquement pas de tableaux de données dans MAP, et quasiment toutes les variables sont locales[30]. d'ou une grande modularité et une grande facilité pour des améliorations et ajouts ultérieurs. De plus, notre application tire pleinement avantage des possibilités d'organisation et de gestion de la mémoire offertes par Windows. Avec le prototype INTERACT, la réservation des emplacements mémoire, une fois les données chargées était permanente. MAP utilise les possibilité offertes par Windows pour transférer sur disque les blocs mémoire momentanément inutilisés, et optimise ainsi l'utilisation du système.

---

[30] C'est à dire que les transferts d'information à l'intérieur du programme se font à travers les paramètres des procédures (passage par paramètres). Ainsi chaque fichier C composant la source du programme est indépendant . La structure de la source C est modulaire.

IV.3.2.2 Organisation des classes de données / opérations de classement

☞ INTERACT

- INTERACT utilise, en mémoire de masse, un seul fichier sous forme ASCII qui contient les observations. Cette caractéristique, comme nous l'avons vu précédemment, se répercute bien sûr en mémoire vive.

- Lors d'un classement, le logiciel manipule directement les blocs mémoire contenant les observations multidimensionnelles pour les réorganiser, et ces mêmes données sont stockées telles quelles lors d'une sauvegarde en mémoire de masse (Cf. Fig.35)

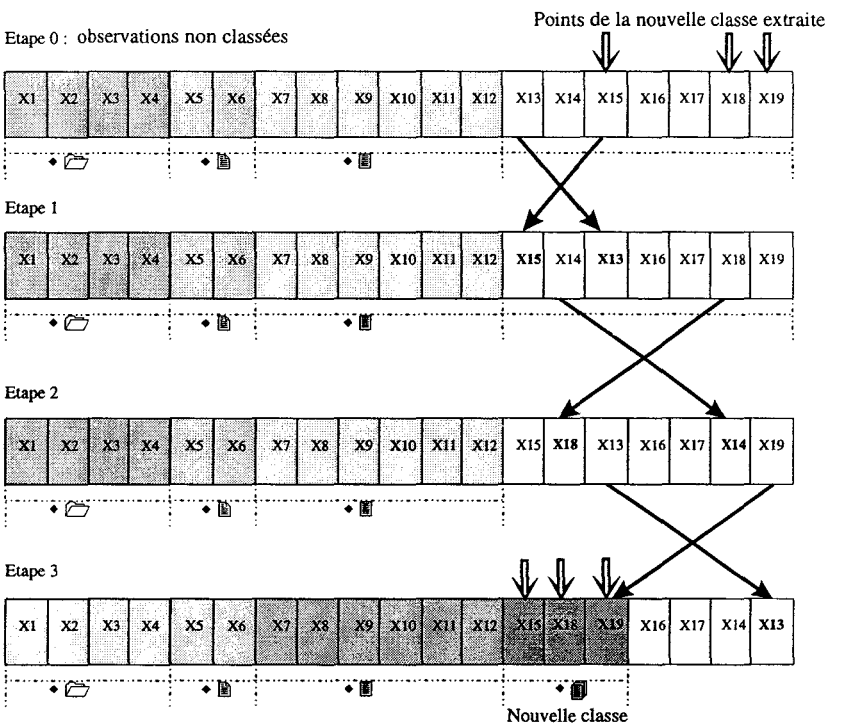


Fig. 35: Procédure de classement du prototype INTERACT

- Ce système est totalement rigide et comporte de plus des inconvénients, puisqu'il a fallu créer un second fichier « trace » pour retrouver à partir de l'indice de l'observation classée, l'indice de cette observation avant classement, afin d'éviter la perte d'information lors du processus de classement. Une solution de toute évidence bancale.

☞ Logiciel MAP

- Contrairement à INTERACT, MAP utilise deux fichiers ASCII au lieu d'un, en mémoire de masse, pour traiter les différentes opérations de classement : Le fichier des observations multidimensionnelles n'est pas modifié[31], bien que l'on puisse bien sûr charger avec MAP un ensemble de données déjà classées pour les afficher ou y effectuer des modifications de classement éventuel. Nous avons veillé à conserver une compatibilité ascendante.

- En mémoire vive, MAP fonctionne avec un bloc mémoire de plus qu'INTERACT : ce sont des variables entières contenant les indices des observations originales. Lorsque l'on effectue une classification sur un ensemble de données, ce ne sont pas les blocs mémoire contenant les données qui sont réorganisés, mais ceux contenant leurs indices (Cf. Fig. 36)

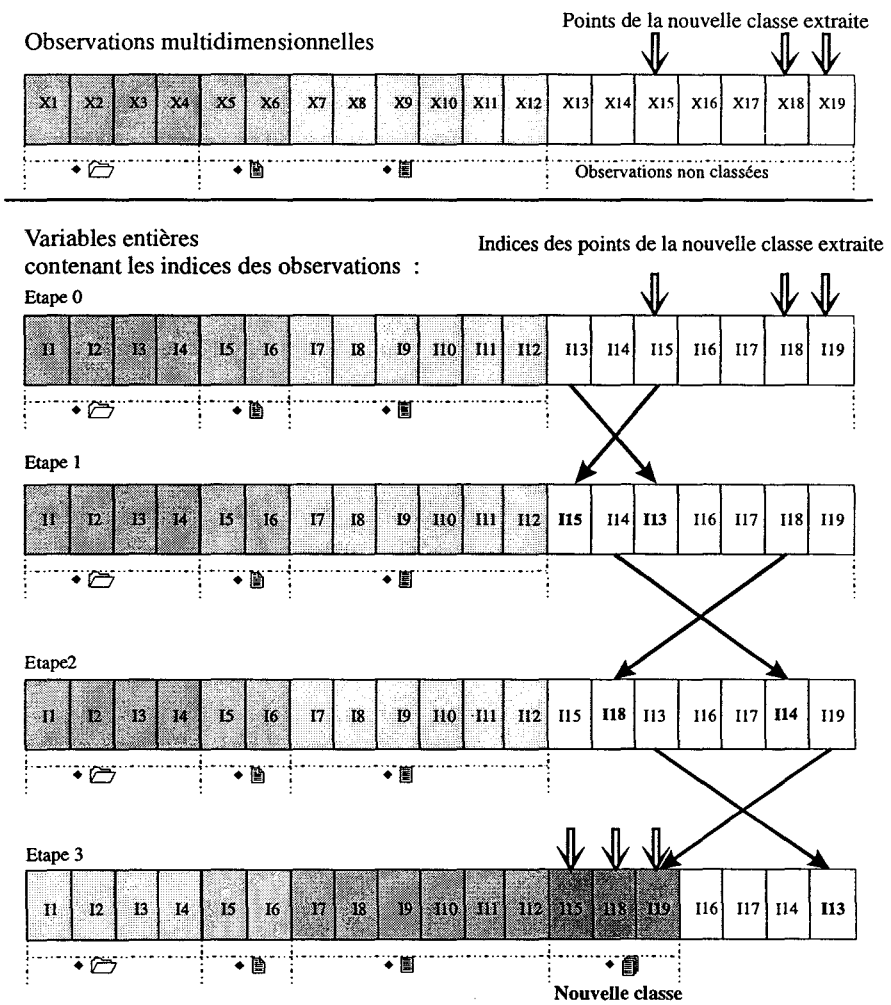


Fig. 36 : Procédure de classement de MAP

[31] L'utilisateur peut cependant choisir de conserver l'ancien format pour sauvegarder le fichier classé

☞ MAP possède, de plus, une souplesse supplémentaire qui a trait à ce qu'on pourrait appeler la « dé-classification » :

• INTERACT, nous permet seulement deux manoeuvres en matière de classification : l'extraction d'une nouvelle classe d'observations, ou bien la destruction d'une classe existante.

• MAP, par contre, permet une manière plus riche d'agencer les observations, puisque l'on peut

- Extraire une classe d'observations

- Fusionner deux classes existantes

• Lorsque l'utilisateur est satisfait de la classification obtenue, il peut alors la sauvegarder en mémoire de masse, cependant ce ne sera pas un nouveau fichier d'observations réorganisé qui viendra écraser le précédent, mais un fichier différent, caractérisé par son extension « .cl » qui sera stocké, et qui contiendra les indices réorganisés. Ainsi on ne modifie en rien le fichier original, et surtout, on peut effectuer plusieurs classifications sur un même ensemble de données, et vérifier ainsi de manière simple leur validité.

☞ **Granularité**

MAP permet de changer la granularité de la représentation (« coarse to fine ») en donnant à l'utilisateur la possibilité d'effectuer une sous classification à l'intérieur d'une classe déjà extraite. Cette manoeuvre ne peut s'effectuer qu'en mode « zoom » c'est à dire lorsque la fenêtre de représentation est cadrée sur la classe que l'on étudie.

IV.3.3 *- Le marquage des points et son utilisation*

♦ Le marquage va permettre à l'utilisateur de désigner l'une des observations affichées à l'écran, et de la mémoriser dans une liste spéciale. Celui-ci peut alors effectuer toute une série de manipulations utiles sur les données en sa possession, afin d'en retirer un maximum d'informations.

♦ On peut marquer une observation de plusieurs manières :

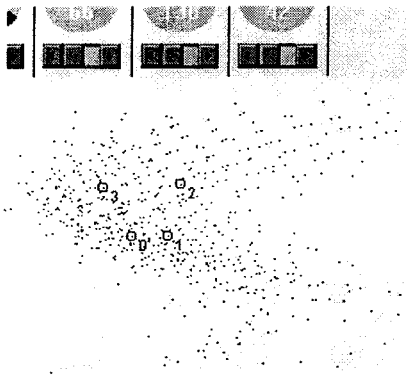


Fig. 37 : points marqués

- hors du contexte de classification, en cliquant deux fois sur sa représentation (Cf. Fig. 37),
- par l'intermédiaire de la boite de dialogue appelée en cliquant « Tags », en entrant l'indice original ou classé de l'observation dans le champ correspondant, à l'intérieur de la boite de dialogue.

IV.3.3.1 La boite de dialogue de gestion des tags

Cette boite de dialogue de gestion des tags est représentée par la figure 38. Passons en revue ses différentes fonctionnalités :

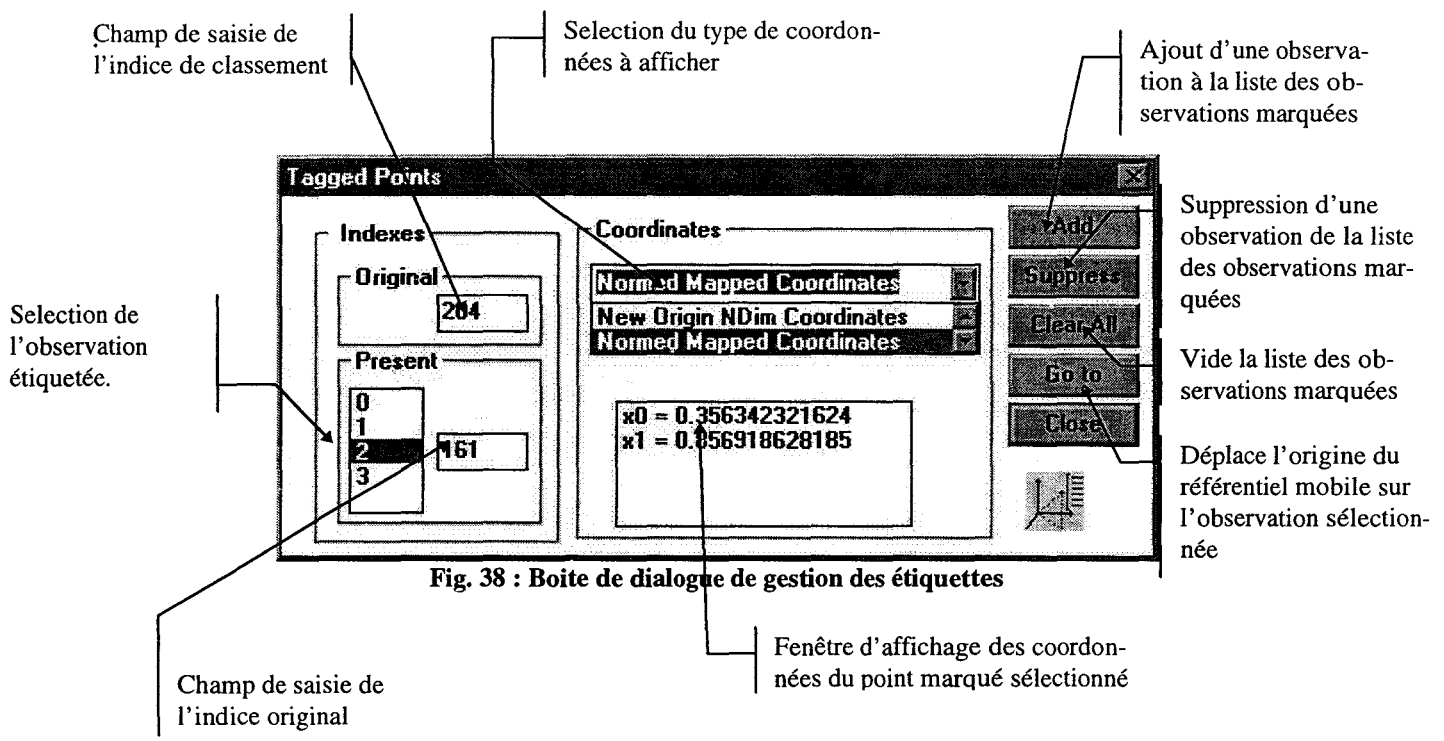


Fig. 38 : Boite de dialogue de gestion des étiquettes

♦A gauche, les deux champs de saisie de l'index de l'observation à étiqueter. Lorsque l'on entre une valeur dans l'un des deux champs de saisie (indice original, ou indice d'ordre dans le classement), l'autre champ, dans le même temps, affiche l'indice correspondant à celui rentré.

♦A droite des champs de saisie des indices se trouvent les coordonnées du point marqué dont l'indice est en surbrillance. On peut choisir d'afficher les coordon-

nées brutes, les coordonnées après changement d'origine, les coordonnées de la représentation, ou bien directement les coordonnées écran du point sélectionné. Les boutons de contrôle sur le côté permettent :

- d'ajouter une observation à la liste des points marqués,
- de retirer une observation de cette liste,
- de vider toute la liste des observations marquées.
- et enfin, de sortir de la boite de dialogue .

Cette boite de dialogue, comme deux autres dans notre application, est *non modale*, ce qui signifie que l'on ne doit pas clore cette fenêtre pour retrouver le contrôle de l'application. Celle-ci fonctionne *en parallèle* avec les autres fonctions de l'application.

Cette nouvelle fonctionnalité offre un grand intérêt, car elle permet une utilisation de notre programme dans un cadre expérimental : par exemple, un botaniste recueille les divers paramètres caractérisant un groupe de plantes, effectue un classement, et désire après coup savoir dans quelle classe se trouve tel ou tel spécimen. Il peut aussi vérifier la validité de l'appartenance d'une observation à une classe en l'étiquetant, et en l'observant lorsqu'il fait évoluer la représentation. Si l'observation a été correctement classée, elle évoluera de manière homogène avec les autres observations de sa classe.

On peut aussi étudier « de plus près » l'ensemble des observations au voisinage d'une observation donnée en marquant celle-ci, puis en déplaçant l'origine du référentiel mobile dessus, c'est à dire en prenant cette observation comme nouvelle origine de notre référentiel mobile. On pourra alors effectuer une sous classification plus fine en bénéficiant de l'effet de loupe que produit cette opération.

Bien sûr, ces nouvelles fonctionnalités tendent à donner à l'utilisateur le contrôle le plus grand possible dans les manipulations qu'il effectue sur les données.

IV.3.4 *L'interface graphique*

- L'interface graphique de notre prototype s'est avérée facile d'accès pour un utilisateur non averti, et les commandes de modification du référentiel mobile faciles à manipuler et suffisamment intuitives. C'est pourquoi lorsque nous avons développé le logiciel final sous Windows, nous avons conservé l'aspect général de l'interface graphique du prototype. L'affichage des points, ainsi que les commandes de manipulation du référentiel mobile et les proportions générales ont été conservés.

Les éléments de l'interface Windows sont présentés dans la figure 39 :

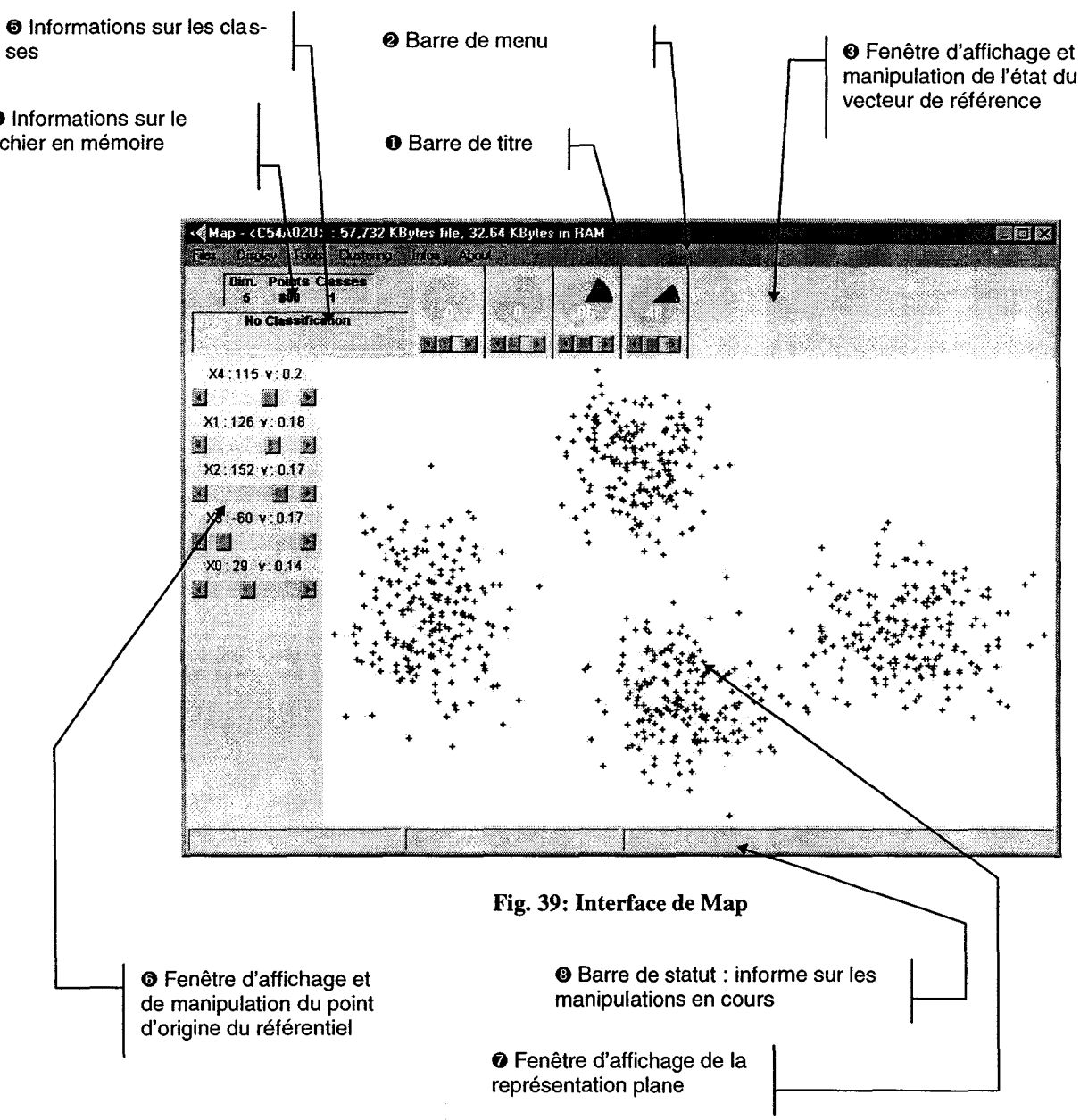


Fig. 39: Interface de Map

Il est à noter toutefois, que si, en apparence, l'application est similaire, la seule composante de programmation du prototype qui a été préservée dans le logiciel final est la partie concernant le calcul des coordonnées scalaires du vecteur de référence à partir de ses coordonnées angulaires. Tout le reste a du être reprogrammé entièrement. En effet, un programme conçu dans l'environnement graphique Windows est par nature orienté objet, ce qui implique une approche de la programmation totalement différente de l'approche procédurale classique.

IV.3.5 *Les fonctionnalités de MAP*

IV.3.5.1 Rappel des éléments innovants par rapport au prototype

On rappelle que MAP comporte, par rapport au prototype un certain nombre de fonctionnalités nouvelles ou améliorées :

- La structure des données en mémoire a été modifiée et permet à présent, d'effectuer des classifications multiples sur un même ensemble de données, stockées chacune dans un fichier propre en mémoire de masse.
- Le gestion du mode de classification donne un contrôle plus complet à l'utilisateur, et une plus grande souplesse de travail
- Lorsqu'une ou plusieurs configurations du référentiel mobile sont appréciées par l'utilisateur, celui-ci peut les stocker, d'abord en mémoire vive, puis en mémoire de masse.
- L'utilisateur peut marquer, étiqueter une observation et ainsi la suivre lorsque celui-ci modifie la représentation en cours (Cf.V-C-3).

IV.3.5.2 Affichage et manipulation de l'état du référentiel mobile

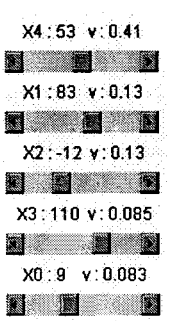


Fig. 40 : Origine du référentiel mobile

L'état du référentiel mobile est affiché et manipulé comme dans le prototype (Cf. Fig. 40). L'affichage est cependant amélioré en ce qui concerne l'origine du référentiel mobile, puisque les axes sont classés par variances décroissantes.

Au dessus de la barre de défilement sont affichés de gauche à droite :

- l'ordre de l'axe,
- la position relative de l'origine,
- la variance du paramètre de l'observation correspondant.

La manipulation de l'une des coordonnées de l'origine se fait par l'intermédiaire de la barre de défilement, et correspond au standard Windows (Flèche de gauche pour diminuer flèche de droite pour augmenter).

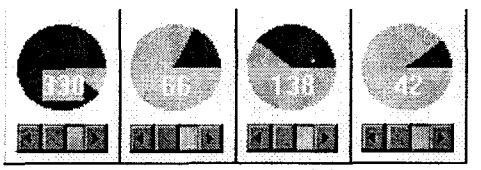


Fig. 41 : Vecteur de référence

L'affichage de la position du vecteur de référence, ainsi que sa manipulation sont identiques à ceux du prototype (Cf. Fig. 41). La seule différence est la possibilité de modifier une coordonnée de l'origine en déplaçant directement le curseur central (amélioration apportée par le standard Windows).

IV.3.5.3 Les fonctions de la barre de menu

IV.3.5.3.a)*Menu « Files »:*

Cette rubrique permet d'accéder aux fonctionnalités proposées en standard dans la plupart des applications Windows (Cf. Fig. 42) : Ouverture d'un fichier de données et sortie de l'application, mais aussi à d'autre rubriques que nous détaillons ci-après :

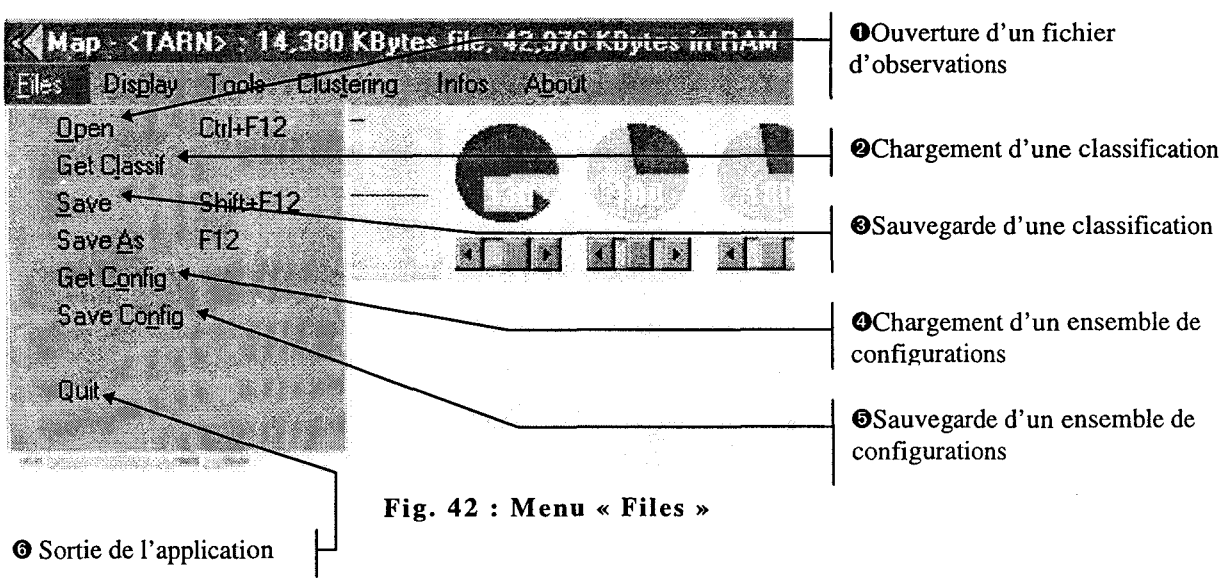


Fig. 42 : Menu « Files »

❶-❷-❸ : Tous les accès au stockage de masse sur le disque dur, que ce soit en chargement, ou en sauvegarde, se font par l'intermédiaire d'une boite de

dialogue standard que possèdent toutes les applications Windows récentes, et que ne n'allons pas détailler ici. Cette boite de dialogue confère un aspect familier à l'interface, dès le premier abord, même pour le novice. MAP comporte cependant une différence dans la structure des données qui rend le chargement et la sauvegarde des données différentes du prototype DOS, cette différence sera détaillée par la suite (Cf. §V.3.6, §V.3.7).

❹-❺ : Chargement et sauvegarde d'un ensemble de configurations ; la gestion des configurations du référentiel mobile sera traitée par la suite (Cf. §V.3.5.3.c).

IV.3.5.3.b)*Menu « Display »*

Cette rubrique permet d'accéder aux différentes caractéristiques de l'affichage :

♦Le sous menu « **Refresh** » permet de choisir le mode de rafraîchissement de la représentation (Cf. Fig. 43) et ouvre trois rubriques:

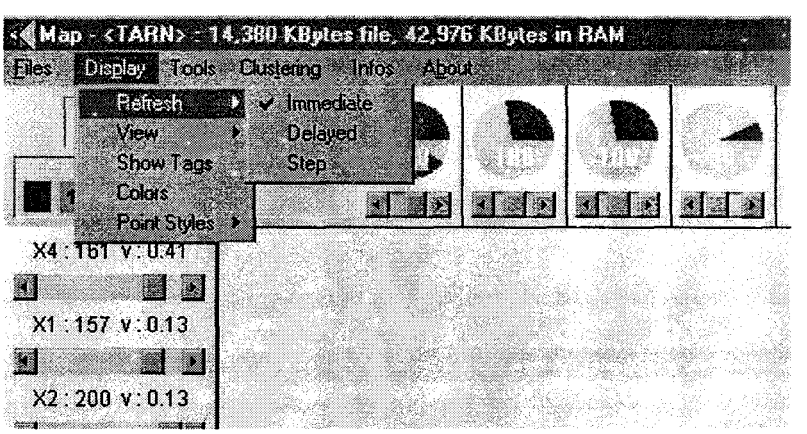


Fig. 43 : Sous-menu « Refresh »

- « **Immediate** » : rafraîchissement continu en temps réel : la représentation est constamment remise à jour tant que l'une des commandes de modification du référentiel mobile est activée[32] .

- « **Delayed** » : Rafraîchissement différé. Le raffraîchissement de la fenêtre de représentation ne se fait que lorsque l'on cesse de modifier la position de l'origine ou la direction du vecteur de référence.

---

[32] c'est à dire tant que l'on continue à modifier la position de l'origine où la direction du vecteur de référence

- « **Step** » ouvre une fenêtre de dialogue modale permettant de modifier les pas d'incrémentation ou de décrémentation des coordonnées du référentiel mobile.

♦ Le sous-menu « **View** » donne à l'utilisateur le choix exclusif entre trois modes de représentation des classes d'observations (Cf. Fig. 44):

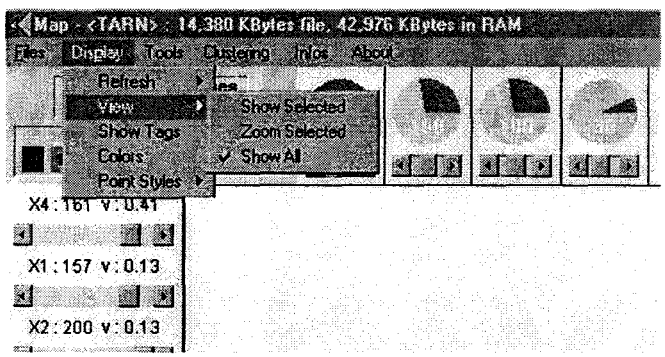


Fig. 44 : Sous-menu « View »

- «**Show Selected** » n'affiche que la ou les classe(s) selectionnée(s) auparavant[33], tout en conservant une fenêtre d'affichage cadrée sur l'ensemble des observations.

- « **Zoom Selected** » n'affiche aussi que la ou les classe(s) selectionnée(s), mais recadre la fenêtre d'affichage sur ces observations seulement.

- « **Show All** » est l'affichage normal de toutes les observations.

♦ La rubrique « **Colors** » ouvre une boite de dialogue modale qui permet de gérer les couleurs des classes ainsi que le fond d'écran : noir ou blanc (Cf. Fig. 45)

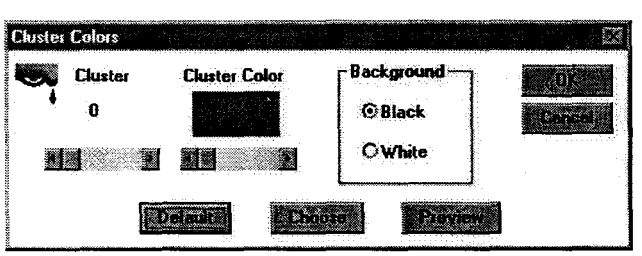


Fig. 45

[33] Nous verrons (Cf. d) que l'on peut sélectionner une ou plusieurs classes à l'aide de la souris.

♦ Le sous menu « **Point Styles** » donne à l'utilisateur le choix entre quatre façon d'afficher les observations : sous forme de points, de croix, de cercles, ou bien de la manière très classique en classification, par le numéro de la classe à laquelle elle appartient (Cf. Fig.46).

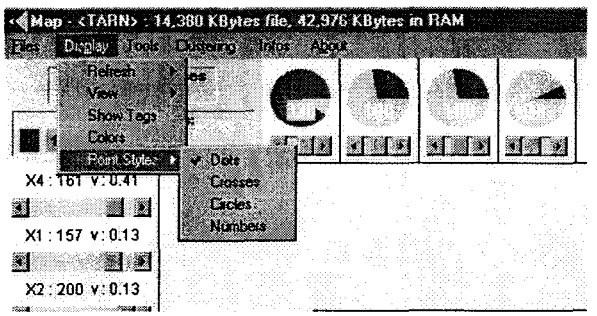


Fig. 46

♦ La rubrique « **Show Tags** » : on a vu précédemment que l'utilisateur pouvait étiqueter les observations dont il désirait connaître la représentation. Lorsqu'elle est validée (cochée), cette rubrique permet de montrer les points qui ont été marqués par l'utilisateur, en faisant apparaître leur ordre d'étiquetage à leur côté.

IV.3.5.3.c)*Menu « Tools »*

Ce menu donne accès aux outils supplémentaires disponibles pour l'analyse d'un ensemble d'observations (Cf. Fig. 47) :

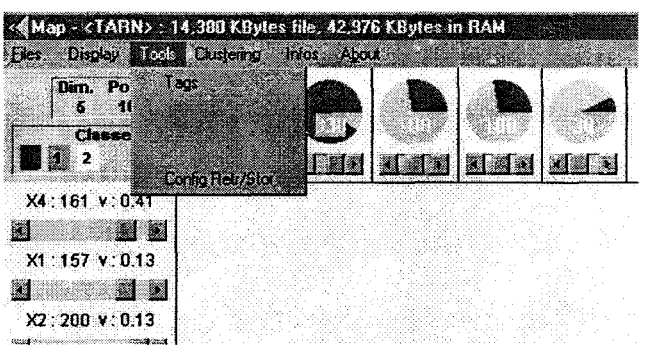


Fig. 47 : Menu « Tools »

♦ « **Tags** » appelle une boite de dialogue non modale qui permet de gérer les points marqués par l'utilisateur (Cf.3).

♦ « **Config Retr/Stor** » appelle une boite de dialogue non modale qui permet à l'utilisateur de sauvegarder, de rappeler et de gérer plusieurs configurations du référentiel mobile (Cf. Fig.48).

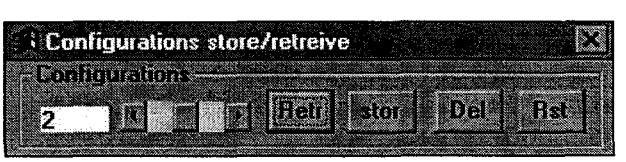


Fig. 48 : Gestion des configurations

Chaque fois que l'utilisateur est parvenu à une représentation des données qui le satisfait, celui-ci peut appeler cette boite de dialogue afin de sauvegarder la configuration présente du référentiel mobile . Il peut ainsi comparer aisément , différentes représentations « parlantes » d'un même ensemble d'observations.

IV.3.5.3.d)*La commande « Clustering » :*

La commande « **Clustering** » appelle, elle aussi, une boite de dialogue non modale, qui permet de gérer avec facilité et souplesse la classification des observations. Les fonctionnalités de cette boite de dialogue son décrites sur la figure 49 :

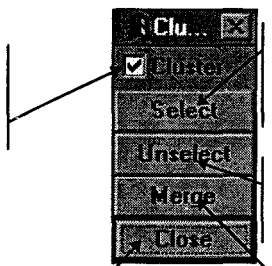


Fig. 49

❶ Les deux modes accessibles de Cluster dans sa case de basculement : *classification / sélection*

❷En mode *sélection,* sélectionne la classe de points dans le voisinage du pointeur

❸En mode *sélection,* dé-sélectionne la classe de points dans le voisinage du pointeur

❹Fusionne les classes sélectionnées

❺Fermeture de la boite de dialogue

Lorsque la boite de dialogue « Clustering » a été activée, les autres fonctions du logiciel MAP sont invalidées et l'utilisateur a le choix entre deux fonctionnements, accessibles par la case *va et vient*[34] « Cluster » (Cf. Fig. 49-❶).

- Par défaut, à l'ouverture de la boite de dialogue, le programme se trouve en mode de *classification*, et la case est, par conséquent validée (cochée).

C'est dans ce mode que s'effectue le création des classes de points. Celle-ci se fait de manière similaire que dans le prototype (Cf.§V.2.2.3). L'utilisateur crée une classe d'observations, en entourant leur représentations par un polygone dont il fixe les sommets à l'aide de la souris :

* En cliquant avec le bouton gauche de la souris, on détermine l'emplacement d'un des sommets. Une croix bleue repère cet emplacement à l'écran.
* En cliquant avec le bouton droit de la souris, on confirme l'emplacement du sommet déterminé auparavant, et la croix le repérant devient rouge.

* En cliquant deux fois avec le bouton gauche, on ferme le polygone dont on vient de fixer les sommets et le logiciel lance la procédure de création de classe (Cf.V-B-2-3). La classe créée est alors réaffichée avec une couleur différente.

- Lorsque l'on dé-valide la case « cluster » en cliquant dessus avec la souris, MAP se trouve en mode *sélection*. Dans ce mode de fonctionnement, l'utilisateur a le choix entre trois actions :

  * Il peut sélectionner une ou **plusieurs** classes en cliquant chaque fois avec la souris sur la représentation d'une des observations appartenant à la classe à sélectionner, puis en cliquant sur le bouton « Select » de la boite de dialogue de classification (Cf. Fig. 49-$\not\subset$).

  * Il peut dé-sélectionner de la même manière une ou plusieurs classes avec le bouton « Unselect » (Cf. Fig. 49-$\subset$).

  * Il peut enfin réunir plusieurs classes ainsi sélectionnées, en cliquant avec la souris sur le bouton « Merge » (Cf. Fig. 49-$\subseteq$)

- Le bouton « Close » permet la fermeture de la boite de dialogue et le retour en mode de fonctionnement normal (Cf. Fig. 49-$\in$)

- Il est à noter que les classes sélectionnées le restent après la fermeture de la boite de dialogue de classification, ce qui permet les affichages de type « Zoom selected» ou « show selected » (Cf. §V.3.5.2.b)

IV.3.5.3.e)*Commande « Infos »*

Cette commande permet d'afficher les informations statistiques relatives aux classes en présence : matrice de covariance de chaque classe, vecteurs moyennes, nombre de points, matrice de covariance globale.

Afin d'illustrer les commandes principales de notre programme dans le cadre d'une utilisation classique, nous présentons à présent un exemple d'utilisation de MAP avec des fichiers d'observations générés artificiellement.

---

[34] « Toggle » en anglais

IV.3.6 *Exemples d'utilisation*

Un exemple de classification type est présenté ici. La démarche de classification est schématisées par un organigramme (Fig. 50) et illustrée par les saisies écran du logiciel correspondant à chaque étape (Fig. 51-55).

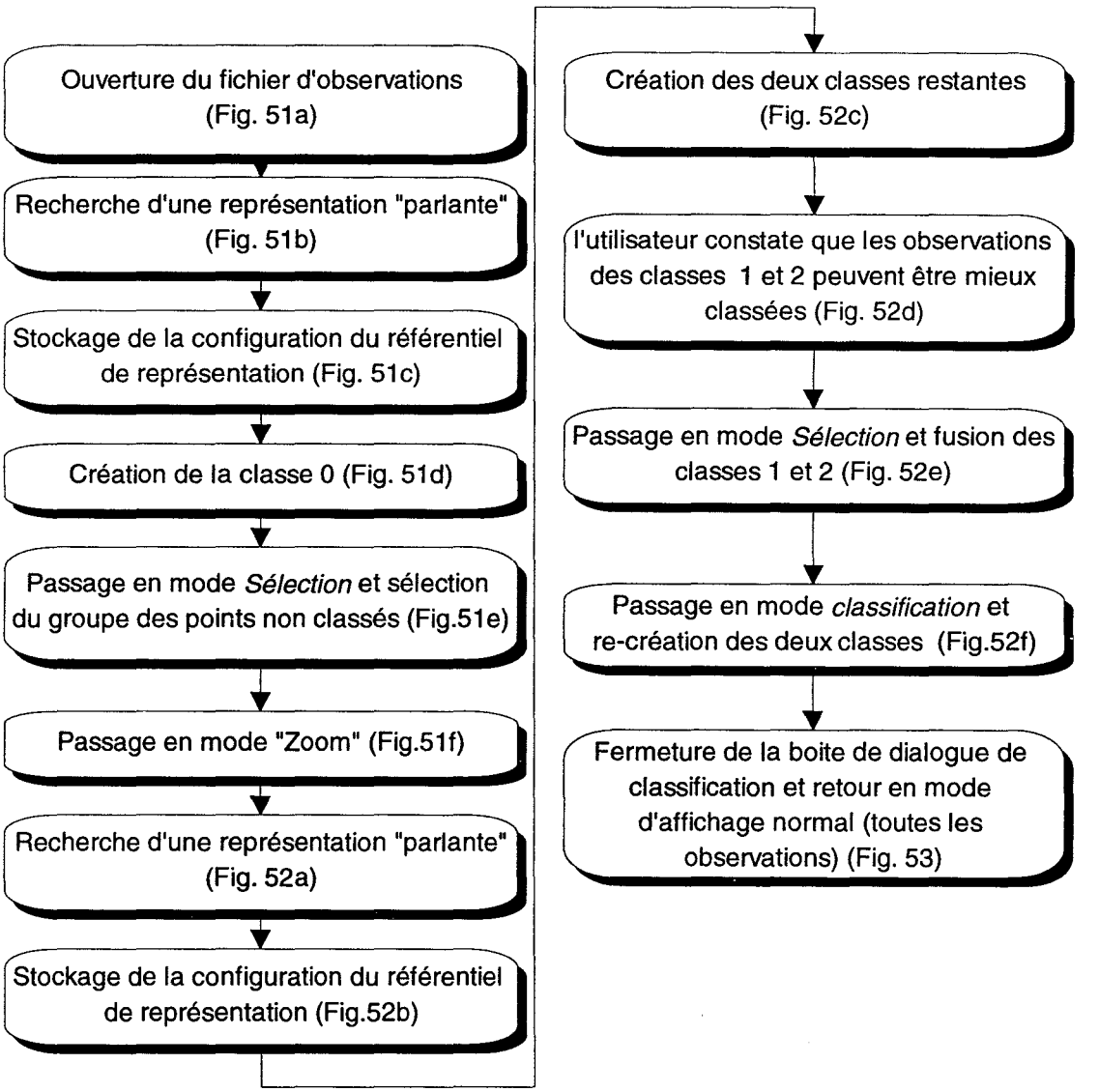
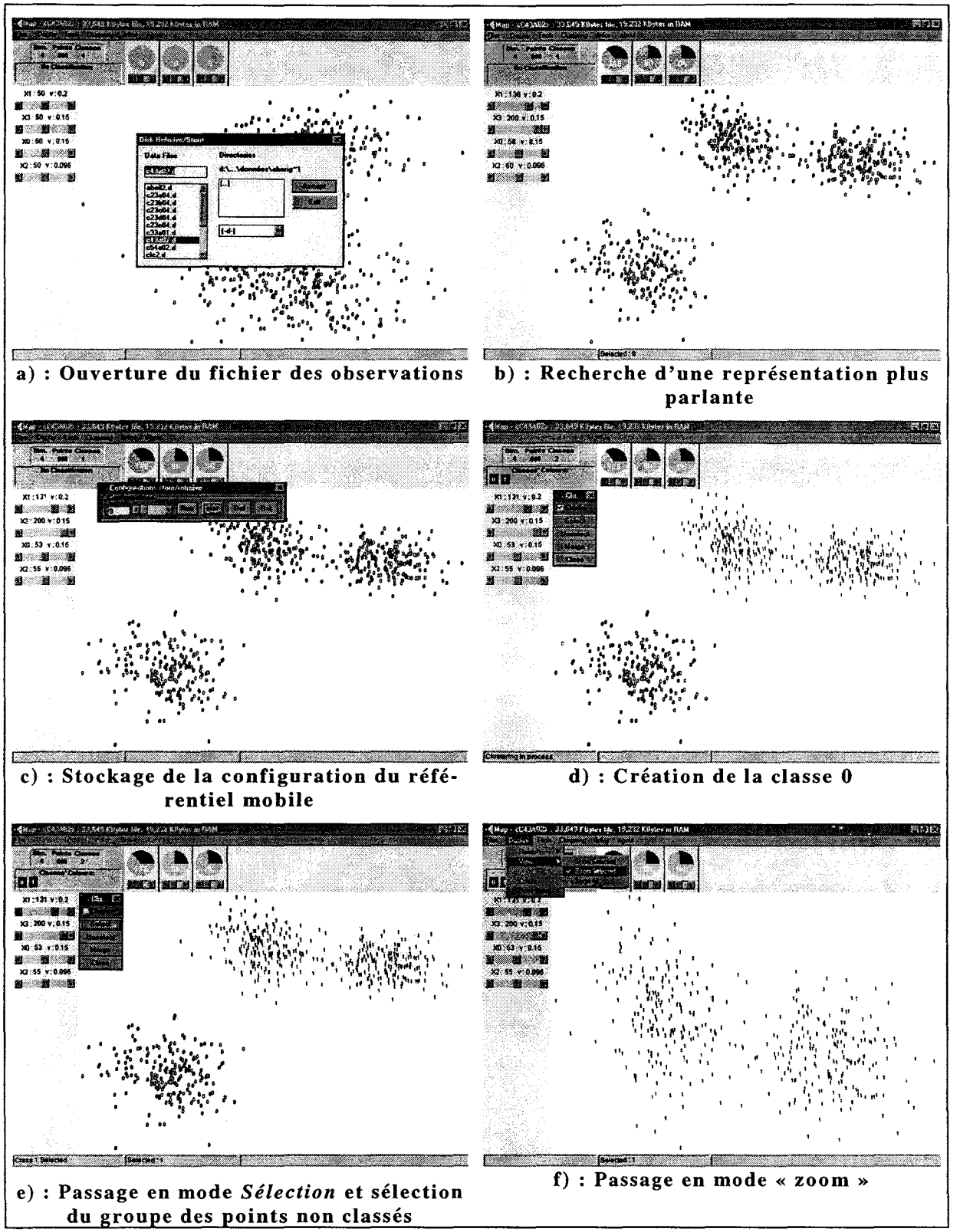


Fig. 50 : exemple de démarche de classification



a) : Ouverture du fichier des observations

b) : Recherche d'une représentation plus parlante

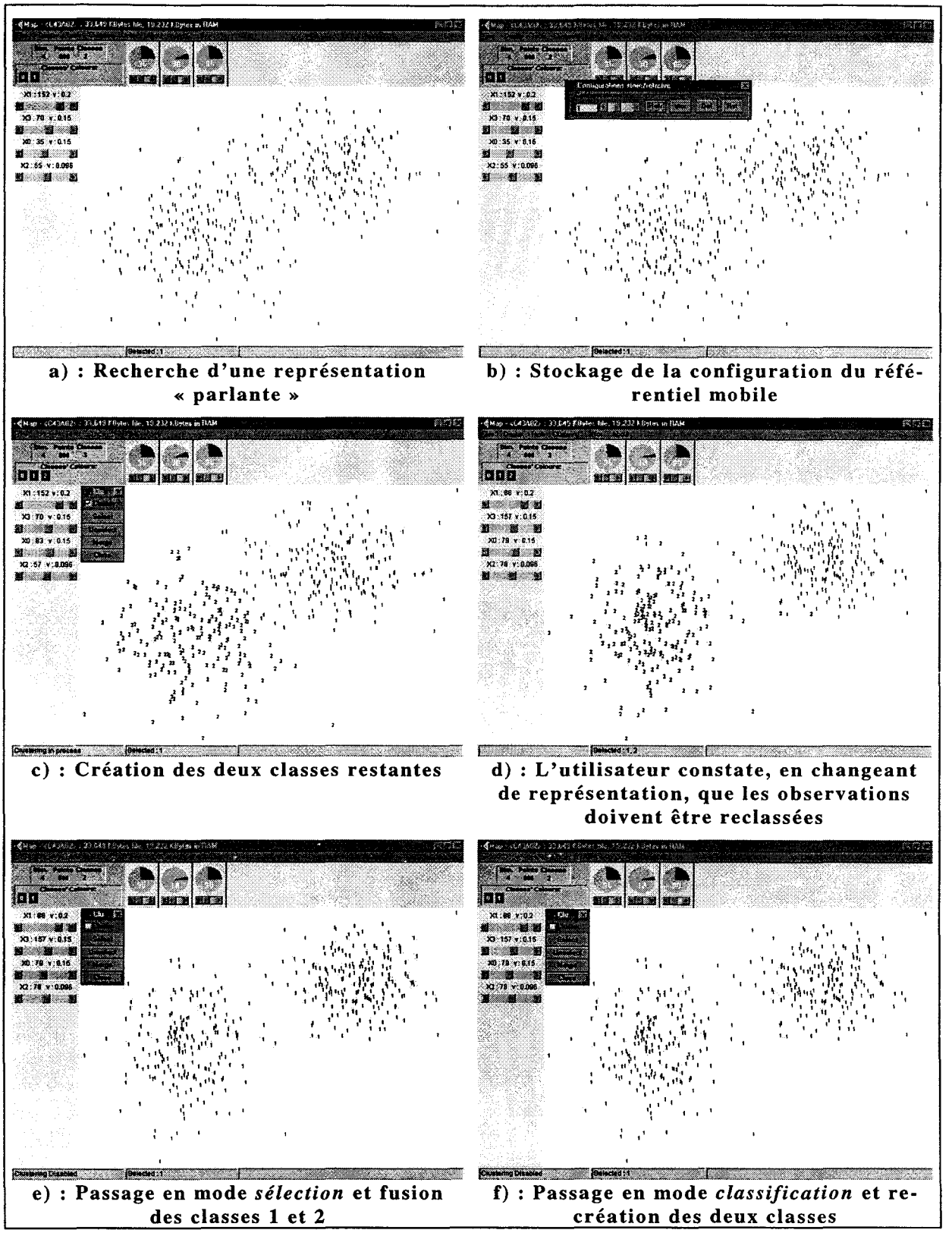
c) : Stockage de la configuration du référentiel mobile

d) : Création de la classe 0

e) : Passage en mode *Sélection* et sélection du groupe des points non classés

f) : Passage en mode « zoom »

Fig. 51 : Etapes d'une démarche type de classification : saisies écran planche 1



a) : Recherche d'une représentation « parlante »

b) : Stockage de la configuration du référentiel mobile

c) : Création des deux classes restantes

d) : L'utilisateur constate, en changeant de représentation, que les observations doivent être reclassées

e) : Passage en mode *sélection* et fusion des classes 1 et 2

f) : Passage en mode *classification* et recréation des deux classes

Fig. 52 : Etapes d'une démarche type de classification : saisies écran planche 2

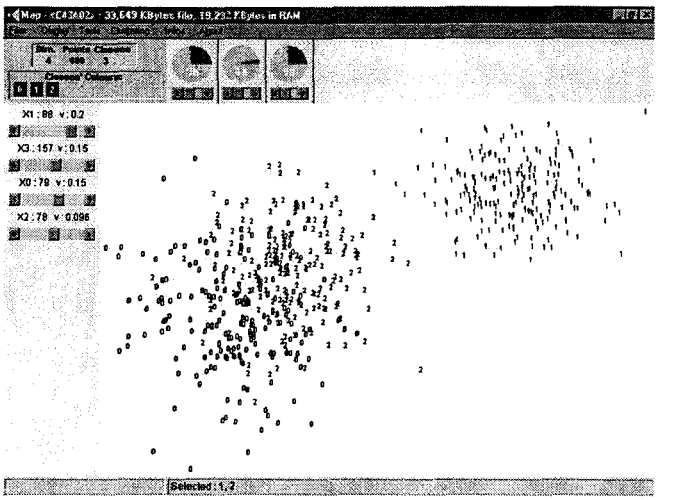


Fig. 53 : Fermeture de la boite de dialogue de classification et retour en mode d'affichage « normal »

La démarche de classification à l'aide de MAP détaillée précédemment est simple. En effet, la combinaison des diverses actions possibles avec Map est bien trop importante pour que nous puissions illustrer ainsi toutes les démarches de classification possibles.

Dans le chapitre qui suit, nous exposons les diverses expérimentations menées sur des fichiers d'observations multidimensionnelles à l'aide soit d'INTERACT , soit de MAP.

CHAPITRE V

EXPERIMENTATIONS

V. EXPERIMENTATIONS

V.1 Sous l'environnement Metawindows

V.1.*1* *Expérimentation sur un fichier de classes générées artificiellement.*

Dans un premier temps, nous avons voulu avoir la possibilité d'étudier l'efficacité de notre logiciel dans le cadre de plusieurs cas types, les uns traitables par les algorithmes de classification automatique, les autres typiquement difficiles à traiter par ces mêmes algorithmes.

C'est pourquoi nous avons, en premier lieu, appliqué notre logiciel à des fichiers d'observations créés à l'aide d'un générateur pseudo-aléatoire de classes gaussiennes multidimensionnelles [KHF86].

L'exemple de représentation qui va suivre est un fichier de données à 5 dimensions, comprenant 4 classes d'observations, que nous avons choisies volontairement proches, et dont l'écart type par paramètre est du même ordre de grandeur que la distance interclasses. C'est un des cas types où des techniques classiques de classification automatique par partitionnement de l'espace de représentation sont mises en difficulté. Les classes gaussiennes créées ont les moyennes et les matrices de covariances suivantes :

- **Classe 1 : 200 points**

$$\text{Moyenne} : X_1 = \begin{bmatrix} 10 \\ 4 \\ 2 \\ 20 \\ 3 \end{bmatrix} \qquad \text{Variance} : [V_1] = 3 \cdot [I_5]$$

- **Classe 2 : 200 points**

$$\text{Moyenne} : X_2 = \begin{bmatrix} 10 \\ 7 \\ 7 \\ 8 \\ 3 \end{bmatrix} \qquad \text{Variance} : [V_2] = 2 \cdot [I_5]$$

- **Classe 3 : 200 points**

$$\text{Moyenne : } X_3 = \begin{bmatrix} 3 \\ 15 \\ 7 \\ 10 \\ 15 \end{bmatrix} \qquad \text{Variance : } [V_3] = 2 \cdot [I_5]$$

- **Classe 4 : 200 points**

$$\text{Moyenne : } X_4 = \begin{bmatrix} 2 \\ 2 \\ 15 \\ 15 \\ 2 \end{bmatrix} \qquad \text{Variance : } [V_4] = 2 \cdot [I_5]$$

où $[I_5]$ est la matrice diagonale unité de dimension 5.

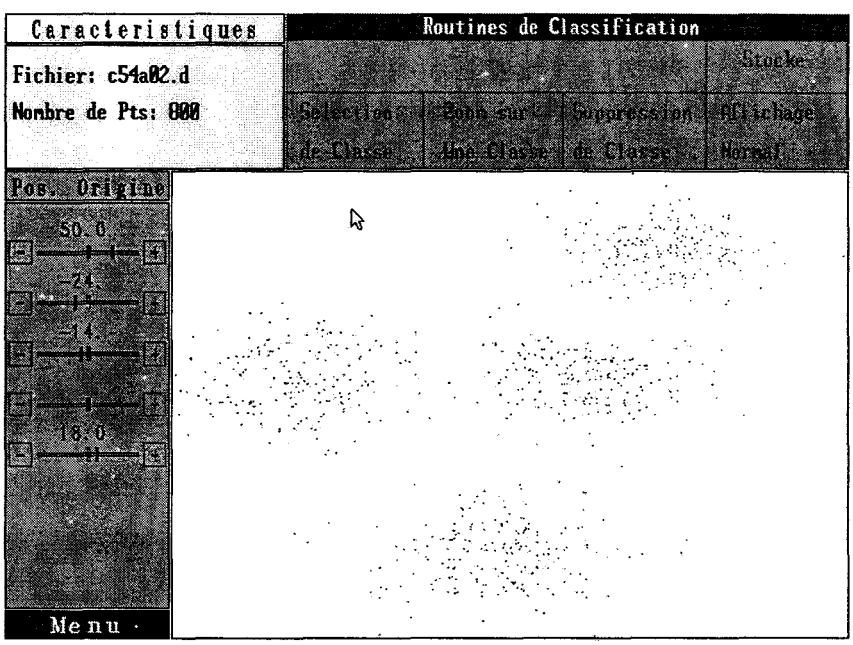


Fig. 54 : Représentation du fichier C54a02.d

On remarque qu'après une recherche de la représentation optimale, c'est-à-dire présentant plusieurs groupes de points distincts, on peut classer sans difficultés les observations représentées, puisque les points images de celles ci forment 4 groupes de points bien séparés, de densité relativement homogène (Fig. 54).

V.1.2 *Données réelles : Analyse d'une image satellite S.P.O.T.*

Toujours dans un but d'expérimentation sur des données réelles, nous avons utilisé notre logiciel pour traiter des images multispectrales en provenance du satellite S.P.O.T. En l'occurrence, il s'agit des images du département du Tarn. Nous avons fait cette étude en collaboration avec Marie-Claire Douchez qui a extrait des images à analyser cinq paramètres sur 256 niveaux de gris chacun : les couleurs rouge (R), infrarouge (Ir), verte (V), la teinte (H) et la saturation (S).

Le nombre d'observations brutes étant cependant trop important (256^2 K octets) pour nous permettre d'utiliser notre logiciel, Marie-Claire Douchez a effectué une réduction de l'information contenue dans les images en utilisant un codage original simple, qui a permis d'obtenir un fichier de 1073 observations à 5 dimensions. Ce codage a été effectué en discrétisant l'espace initial des observations de dimension 5, en h^5 hypercubes [POV82]. Nous avons pris comme nouvel ensemble des observations, l'ensemble des centres des hypercubes contenant au moins une observation originale. Nous avons alors fait varier le pas de discrétisation h de l'espace d'origine jusqu'à obtenir un nombre d'observations utilisables par notre prototype INTERACT.

On remarque sur la représentation (Fig. 55) 3 groupes de points clairement séparés. L'un étant plus facilement isolable que les deux autres (en bas à gauche sur la figure 55). Nous avons donc, à partir de cette représentation, isolé 3 classes d'observations.

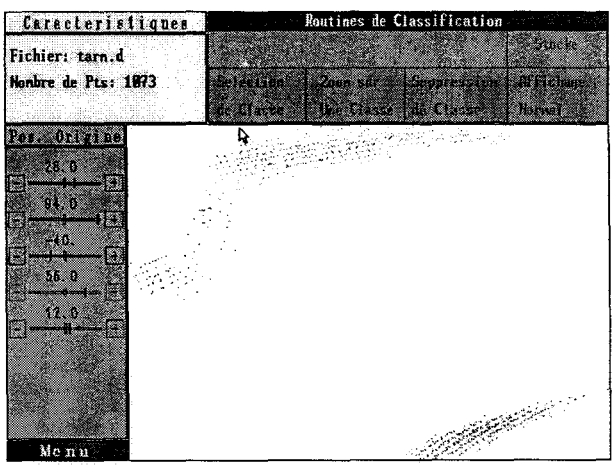


Fig. 55: Représentation du Fichier d'observations multidimensionnelles extraites des images satellite du Tarn

Un problème se posait alors à nous : comment mettre à profit les informations apportées par notre classification, étant donné que nous n'avions pas pu effectuer cette dernière sur la totalité des observations multidimensionnelles tirées des images satellites.

Nous avons résolu ce problème en utilisant notre classification pour effectuer une segmentation sur images en niveaux de gris. Nous avons donc choisi d'extraire les maxima et minima de chacune des cinq composantes, et ceci pour chaque classe.

Nous avons retrouvé les extrema réels à partir des coordonnées normalisées et des latitudes d'évolution L_i de chaque paramètre (Fig. 56) (Cf. §V.2).

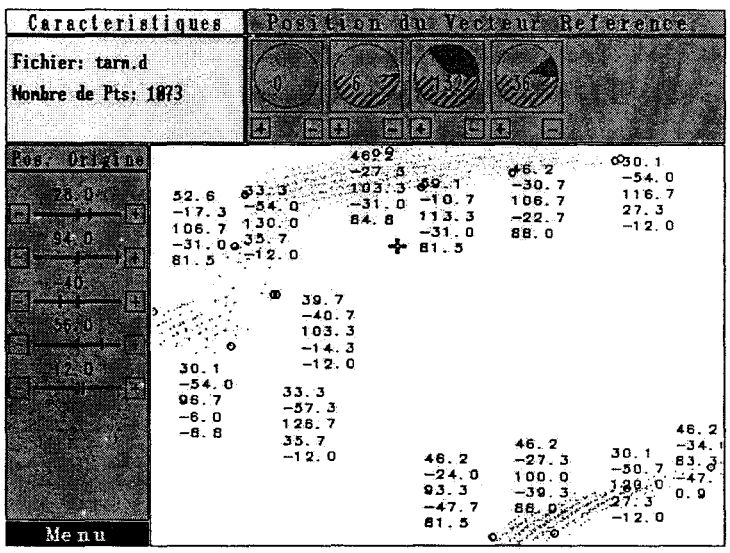


Fig. 56 : Visualisation des extrema

Comme on pourra le constater dans les photographies qui suivent, une comparaison entre les images brutes en niveaux de gris des trois composantes et ces mêmes images, segmentées à partir de notre classification, montre que les trois classes isolées correspondent, pour chaque composante, à des portions de terrain différentiables sur la photographie en niveau de gris.

Ces résultats probants ouvrent le champ à un nouveau moyen d'étude des images satellites.

SPOT : Images satellites multispectrale du Tarn en niveaux de gris avant et après segmentation.

Photo 1 : Composante rouge originale

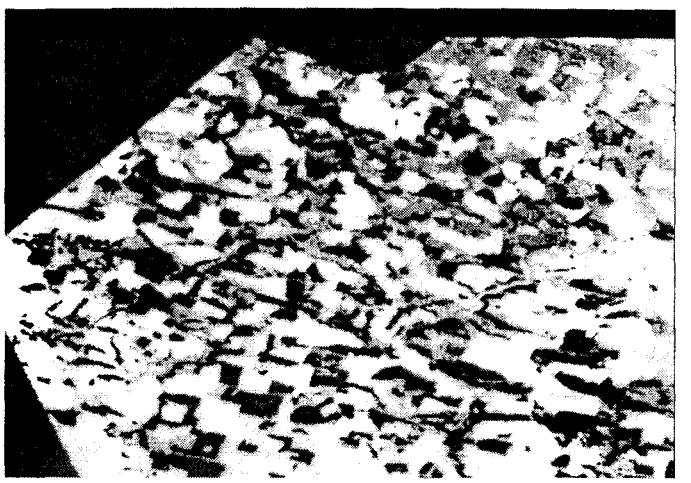


Photo 2 : Composante rouge segmentée

Photo 3: Composante verte originale

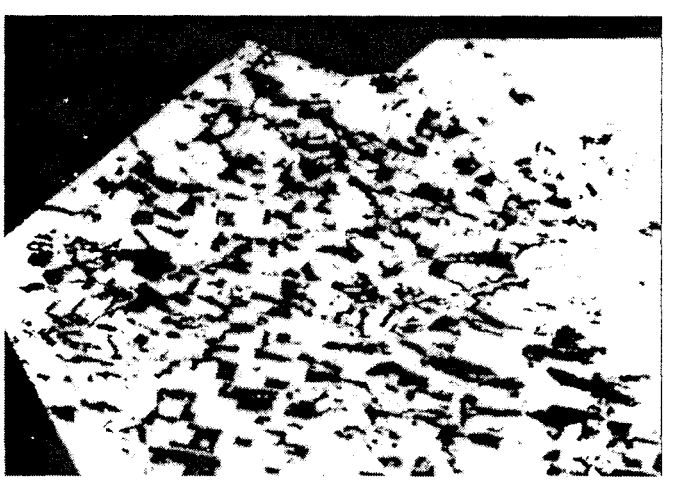


Photo 4 : Composante Verte segmentée

Photo 5 : Composante infrarouge originale

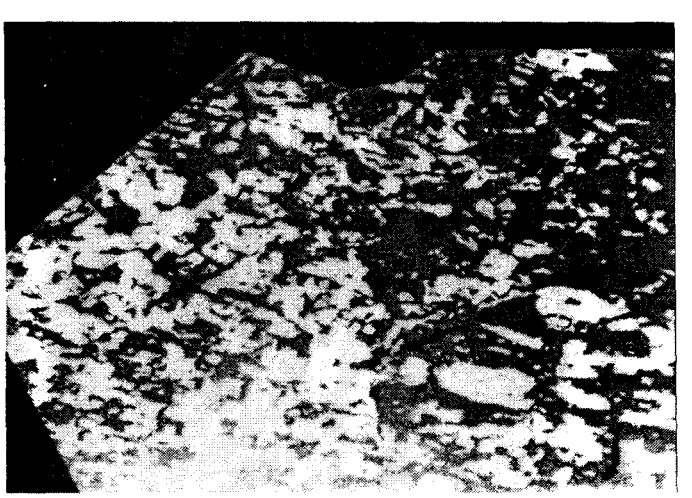


Photo 6 : Composante infrarouge segmentée

V.2 Sous l'environnement Windows

V.2.1 *Récapitulatif*

Les divers exemples exposés précédemment, en confirmant la validité de notre approche, nous ont encouragés à poursuivre dans cette direction et à améliorer le logiciel qui en découlait.

Nous exposons à présent la suite des expériences menées à l'aide du nouveau logiciel sous Windows, à la fois plus souple et plus puissant que la version d'essai sous Metawindows

V.2.2 *Etude d'un fichier d'observations apidologiques*

Nous avons analysé un fichier d'observations biométriques étudiées par Hadria Fizazi à l'occasion de sa thèse de Doctorat [FIZ87]. Ces observations concernent un échantillon de 1302 abeilles provenant de différentes îles des Antilles Françaises. Six paramètres ont été mesurés sur chacune des abeilles [FRE81], à savoir :

- la coloration (Longueur de la bande jaune sur le tergite n° 2)
- la pilosité (Longueur des poils du tergite n° 5)
- le tomentum (Longueur de la bande du tomentum du tergite N° 4)
- la longueur de la langue (Proboscis)
- l'index cubital A (Longueur des veines A de l'index cubital)
- l'index cubital B (Longueur des veines B de l'index cubital)

V.2.2.1 Représentation à l'aide de MAP

Dès la représentation initiale affichée par notre logiciel (Etat par défaut du référentiel mobile lors de l'ouverture d'un fichier d'observations, origine du référentiel mobile au centre des observations) , ce fichier est apparu comme composé de deux classes bien distinctes, de densités, de dispersions et de moyennes différentes.

Cette représentation est exposée Figure 57.

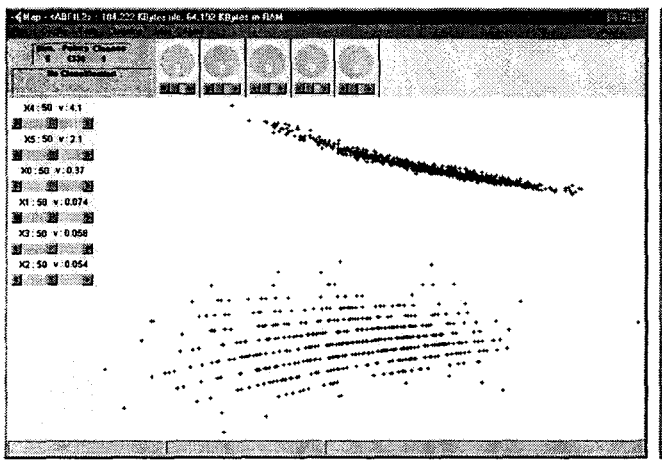


Fig. 57 : Observations apidologiques : représentation initiale

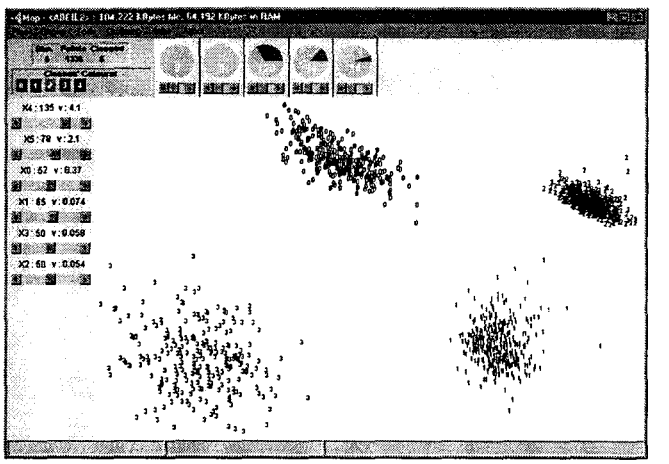


Fig. 58 : Observations apidologiques classées

La recherche d'une meilleure représentation nous a permis d'isoler quatre classes d'observations bien distinctes comme on peut le voir sur la figure 58.

C'est là que nous avons fait appel à une approche qui augmente de beaucoup les capacités de notre logiciel, et qui est la démarche « Coarse to fine ».

La démarche « coarse to fine » a permis l'étude séparée de chaque classe précédemment isolée comme s'il s'agissait, à chaque fois, d'un ensemble d'observations différent (Cf. §V.3). Par le biais de cette démarche, nous avons pu isoler une sous-classe, dans chacune des classes 1 et 2, mettant ainsi en évidence 6 classes d'observations (Fig. 57 à 63) :

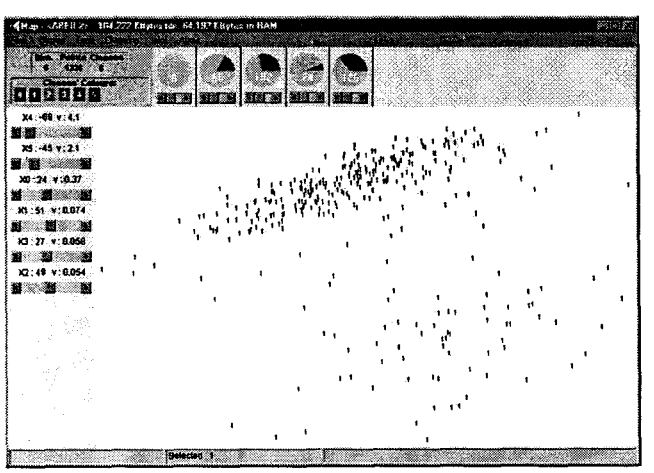


Fig. 59 : Mode Zoom sur la classe 1

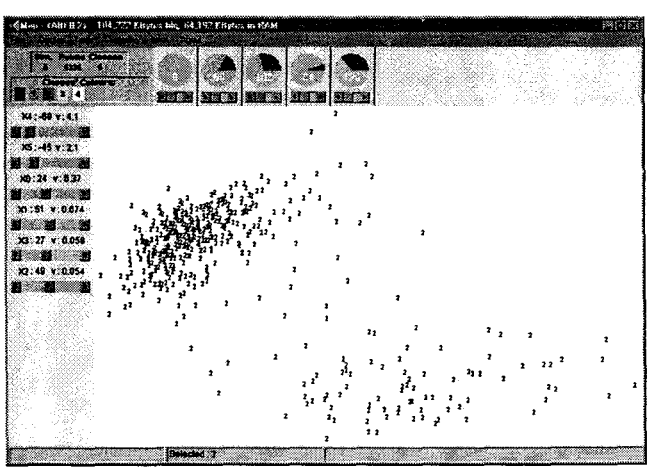


Fig. 60 : Mode Zoom sur la classe 2

Les figures 61 et 62 montrent les deux sous-classes (classe5 et 6) après classification

On remarque sur la représentation initiale que les deux sous-classes n'auraient pas pu être isolées sans l'approche « coarse to fine » (Fig. 63).

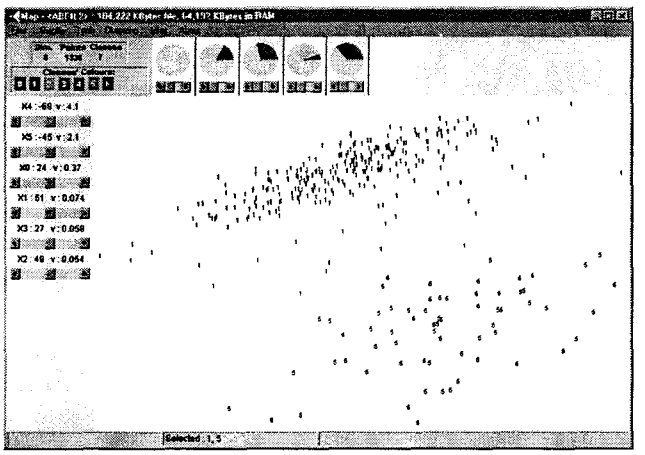


Fig. 62 : Isolation d'une sous classe à la classe1

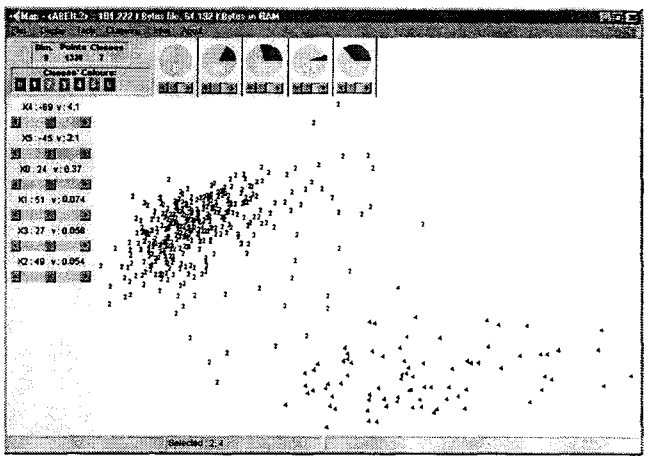


Fig. 61 : Isolation d'une sous-classe à la classe 2

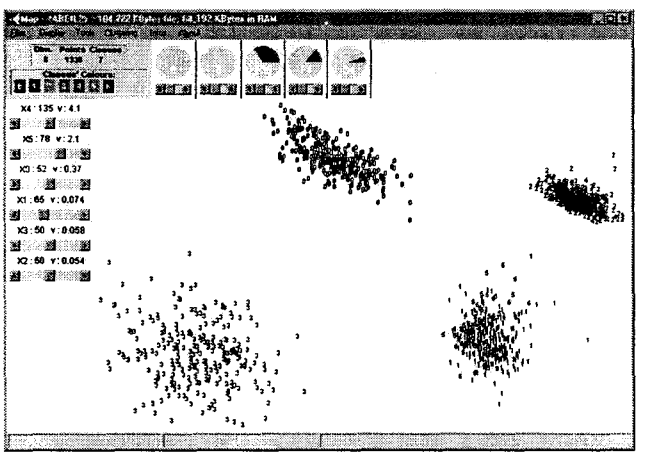


Fig. 63 :Représentation originale 6 classes d'observations

Il nous a paru intéressant de comparer nos résultats avec ceux obtenus par l'intermédiaire d'autres types de représentations. C'est ce que nous traitons dans la partie qui suit.

V.2.2.2 Autres représentations

Ces représentations ont été obtenues par Denis Hamad, Stéphane Delsert et Mohamed Bétrouni, dans le cadre de leur recherche sur les représentations planes par l'intermédiaire de réseaux de neurones (II-C-2-b)[BDH95] :

On remarque que la représentation par Analyse en Composante Principale (II-C-I-a) ne nous permet d'isoler que deux groupes d'observations, assez difficiles à différencier dans les données apidologiques (Fig.64).

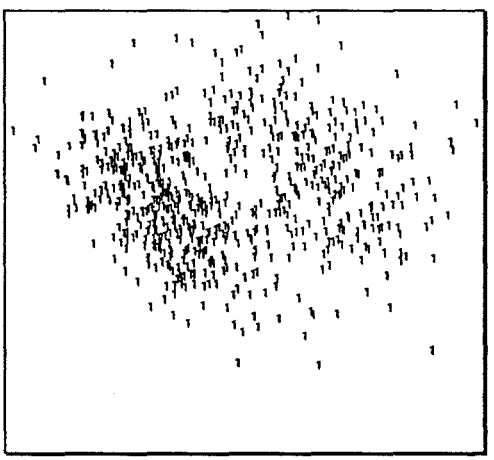


Fig. 64 : Représentation des observations apidologiques par Analyse en Composantes Principales

La représentation par l'algorithme de Sammon (Cf. Fig. 65) fournit un résultat assez similaire au notre et permet bien d'isoler quatre classes principales et deux sous classes.

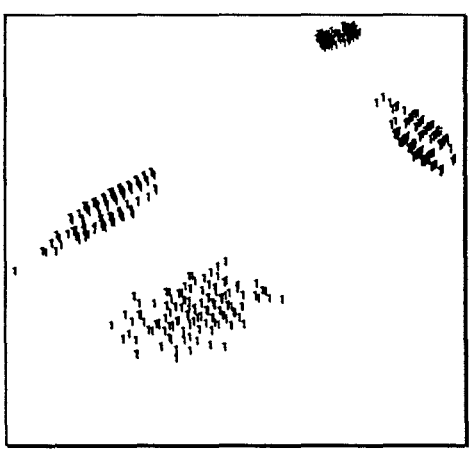


Fig. 65 : Représentation des observations apidologiques par l'algorithme de Sammon

La dernière représentation (Fig. 66) est obtenue à l'aide d'un réseau de neurone à 5 couches cachées, minimisant le critère de Sammon [BDH95] .

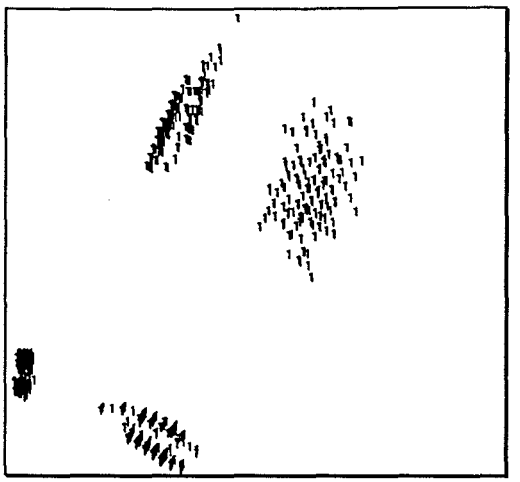


Fig. 66 : Représentation des données apidologiques par implantation neuronale de l'algorithme de Sammon

Ces résultats sont tous conformes aux nôtres et mettent en évidence la validité de nos modes respectifs de représentation.

V.2.3 *Etude de l'ensemble d'observations « Iris Data »*

C'est un ensemble classique d'observations utilisé par Fisher [FIS70], ainsi que par la plupart des chercheurs en analyse de données, pour tester les procédures de classification automatique.

Cet ensemble est formé de mesures faites sur 50 spécimens d'iris provenant de trois espèces différentes.

V.2.3.1 Représentation et classification à l'aide de MAP

Notre logiciel permet à l'utilisateur de sauver en mémoire de masse autant de classifications voulues qu'il le désire sur un même fichier (Cf. §V.3.2.2). Les observations du fichier des iris ayant été saisies espèce après espèce, on possède d'emblée les informations concernant les trois classes d'iris en présence. Nous avons donc sauvegardé cette « classification » d'origine, puis nous avons repris l'étude de ce fichier après avoir supprimé celle-ci et avons entrepris de le classer nous même, afin de tester une fois de plus la validité de notre méthode. Les figures suivantes montrent un résumé des étapes successives de notre démarche de classification.

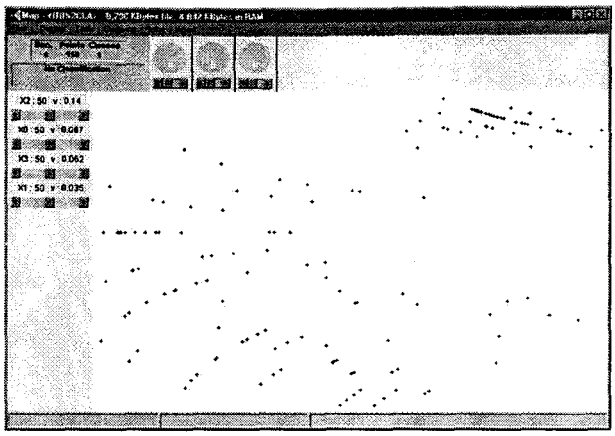


Fig. 67 : Iris Data, représentation initiale, obsevations non classées

La première configuration présentée Fig. 67 est la configuration par défaut à l'ouverture du fichier. Avec cette représentation, il est déjà possible d'isoler une classe qui se distingue sans peine du reste des données. Nous avons cependant cherché une représentation meilleure (Fig. 68)

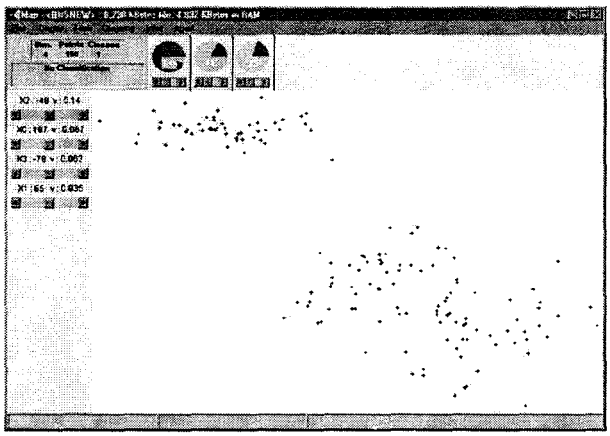


Fig. 68 : Iris Data, observations non classées, 2ème configuration du référentiel mobile

A partir de la seconde configuration du référentiel mobile présentée Fig. 68, nous avons pu isoler la première classe d'observations, soit la classe 0. La figure 69 montre la représentation des observations après création de la classe 0. La « classe » 1 est par définition l'ensemble des observations non classées (Cf. §V.3.2.2).

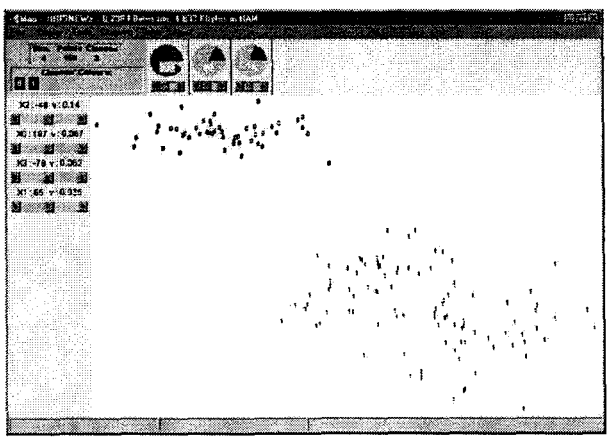


Fig. 69 : Iris Data, création de la classe 0

La classe 0 une fois crée, nous avons procédé par une démarche « coarse to fine » pour isoler les deux classes restantes, en effectuant un *zoom* sur le reste des données (Fig. 70).

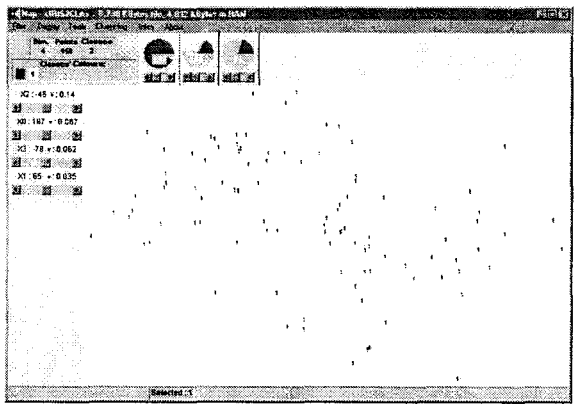


Fig. 70 : Iris Data représentation des 100 observations non encore classées

Nous avons par la suite recherché une nouvelle configuration du référentiel mobile qui permette une meilleure représentation des deux classes restantes en mode « zoom », afin de pouvoir les classer de manière optimale (Fig.71).

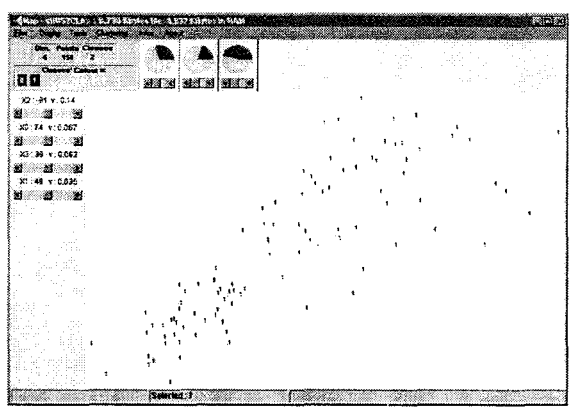


Fig. 71 : Iris Data, données non classées, 3ème configuration du référentiel mobile

Cette troisième configuration du référentiel mobile, nous donne accès à une représentation des observations restantes où apparaissent nettement deux groupes. Nous avons donc isolé ces deux classes d'observations (Fig. 72).

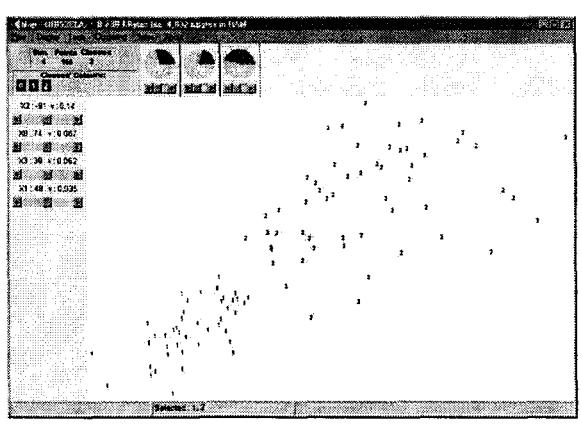


Fig. 72 : Iris Data, 3ème configuration du référentiel mobile classes 1 et 2

Notre fichier d'observations est à présent classé. Il nous reste à comparer notre classification avec l'organisation originale des observations. Nous avons donc récupéré celle-ci, et avons comparé les deux, en gardant la même configuration du référentiel mobile (Fig. 73, 74).

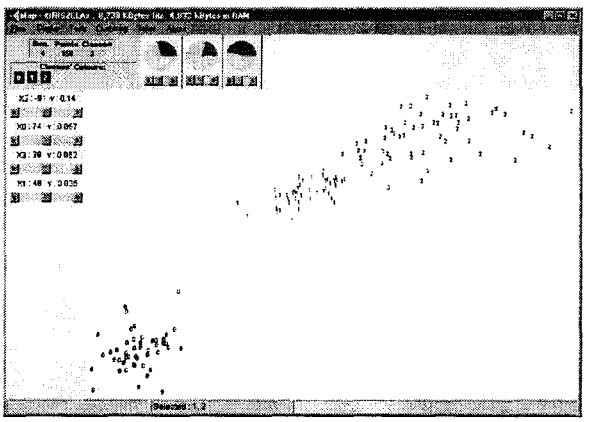


Fig. 73 : Iris Data 3ème configuration du référentiel mobile, notre classification

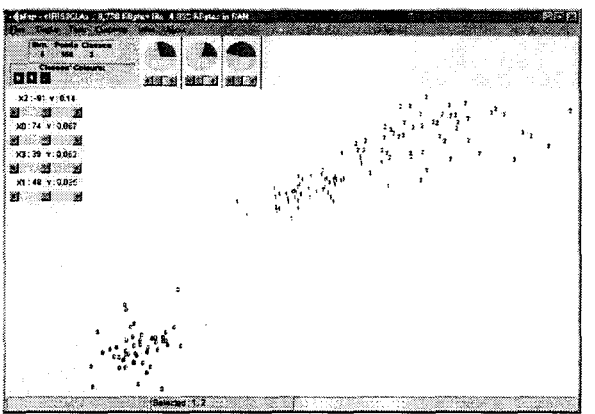


Fig. 74 : Iris Data, 3ème configuration du référentiel mobile, organisation originale

On peut constater que la classe 0 a été correctement isolée, mais que des différences existent entre les deux dernières classes d'observations. Comparons en mode *zoom* notre classification avec la classification originale, en gardant la même configuration du référentiel mobile (Fig. 75)

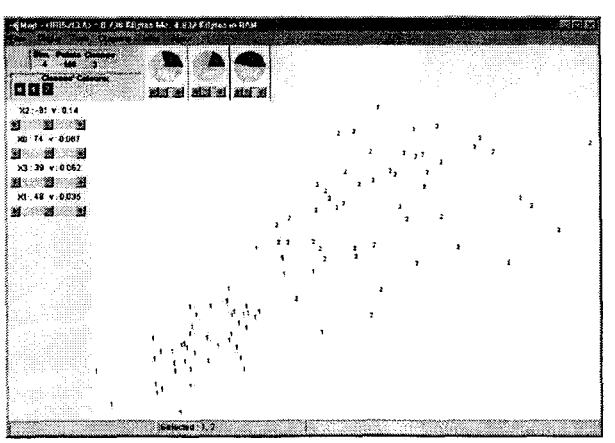


Fig. 75 : Iris Data, 3ème configuration du référentiel mobile, Classes originales 1 et 2 en mode Zoom

On remarque que six observations appartenant à la classe 1 ont été attribuées à la classe 2, ce qui constitue donc une erreur de 6/150, soit un taux d'erreur de 4 % ce qui représente un bon résultat.

V.2.3.2 Autres représentations

Les trois figures suivantes montrent les représentations de cet ensemble d'observations obtenues par J. Friedman et J. Tukey avec leur algorithme de projections révélatrices [FTU74]: la figure 76 est une représentation globale obtenue en prenant les deux axes principaux les plus importants comme point de départ. Les figures 77et 78 reprennent les 100 observation restantes, après séparation des 50 observations qui formaient une classe nettement séparée dans la première représentation.

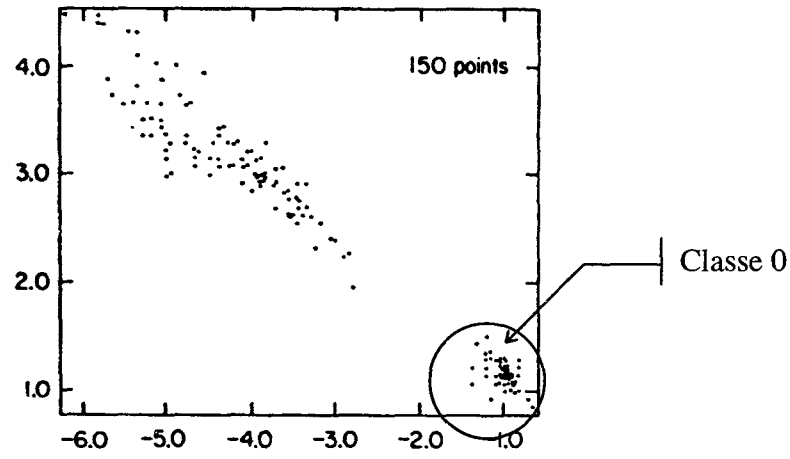


Fig. 76 : Iris Data, représentation par projections Révélatrices N°1

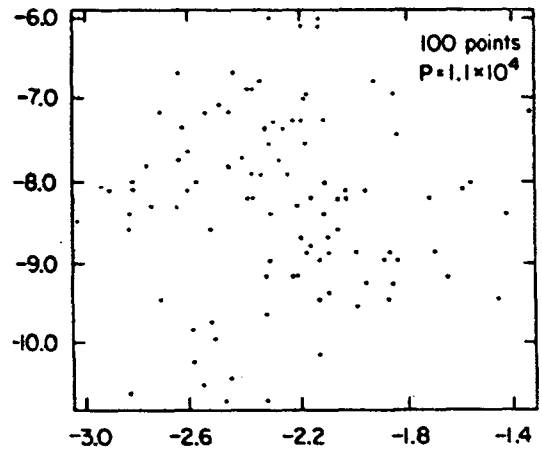


Fig. 77 : Iris Data, sans la classe 0, représentation par ACP

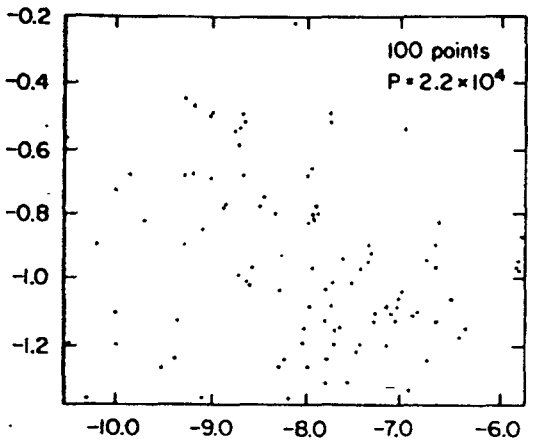


Fig. 78 : Iris Data, sans la classe 0, représentation par Projections révélatrices obtenue en partant de fig. 76

La figure 79 nous montre une représentation d'Iris Data obtenue par M. Bétrouni et Denis Hamad à l'aide d'un réseau de neurones « feedforward » multicouches optimisant le critère de Sammon.

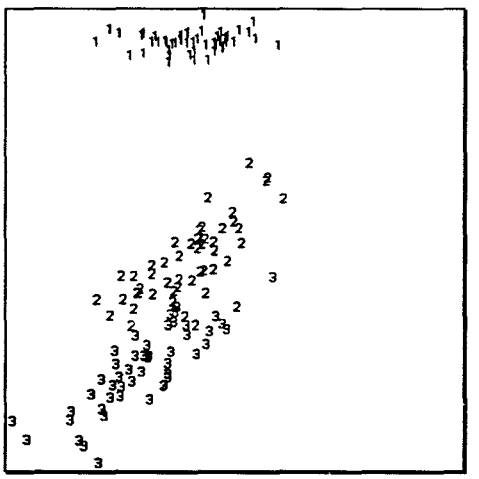


Fig. 79 : Représentation d'Iris Data par un réseau de neurones optimisant le critère de Sammon

Sur cette dernière représentation, les classes 0, 1 et 2 dans notre notation, sont notées classes 1 , 2, et 3.

V.2.3.3 Conclusion

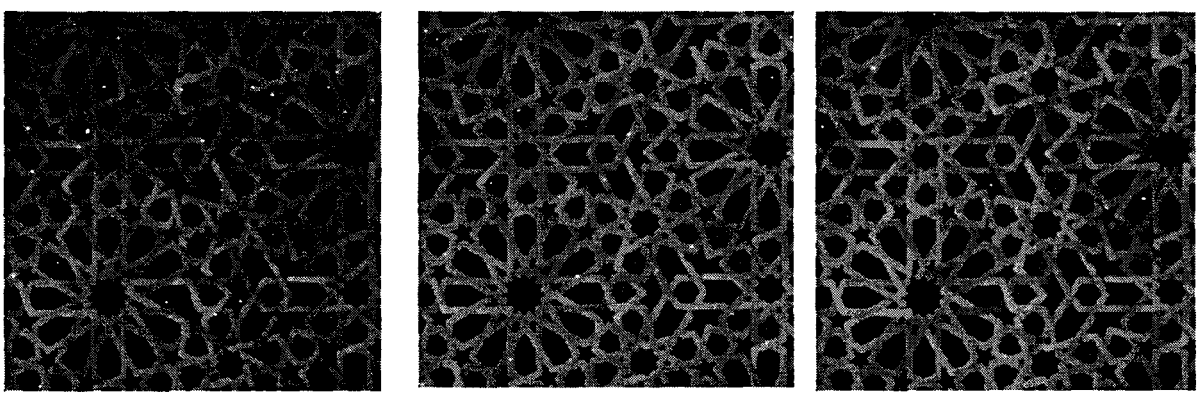
En comparant nos représentations avec les autres représentations disponibles, nous pouvons constater que MAP nous a fourni la meilleure représentation de ce fichier d'observations (c'est-à-dire la représentation présentant la meilleure séparation des trois classes existantes). Les classes 1 et 2 sont, dans les deux cas, identifiables, mais pas aussi clairement séparées qu'avec MAP. De plus, MAP nous a permis de classer les observations avec un taux d'erreur particulièrement faible puisque de 4%.

V.2.4 *Segmentation d'Images Couleurs par Classification Interactive*

Cette expérimentation a été menée en collaboration avec Vincent Ultré dans le cadre de sa thèse [UFE95][UTE95][UTR96] et se compose de deux parties. La première concerne l'analyse, à l'aide de MAP, du contenu chromatique d'images de mosaïques traditionnelles polychromes de la ville de Fès au Maroc, en vue de leur archivage et de leur restauration. Il s'agit de traiter ces dernières à l'aide de procédés permettant d'éliminer les points peu significatifs du point de vue chromatique, jusqu'à obtention d'une segmentation finale de l'image couleur. On obtient ainsi des classes de pixels homogènes vis à vis des composantes R,V,B, qui, dans le cas de la mosaïque, correspondent aux différents matériaux utilisés. La deuxième partie traite de la segmentation interactive directe d'images brutes à l'aide du logiciel de classification interactive.

V.2.4.1 Utilisation du logiciel pour l'aide à la segmentation

Nous traitons des images de mosaïques en couleur. Celles-ci comportent cinq couleurs, noir, blanc, rouge, vert et bleu. Nous savons donc que l'image doit contenir cinq classes de points. Deux classes correspondent aux pixels achromes (la classe des points blancs et la classe des points noirs) et les trois autres aux pixels colorés, c'est-à-dire les points rouges, verts et bleus. La figure 80 représente les trois plans rouge, vert et bleu de cette image.



Plan Rouge Plan Vert Plan Bleu

Fig. 80 :Les différents plans de l'image mosaïque.

La figure 81 montre la meilleure représentation des points dans l'espace R.V.B. obtenue à l'aide du logiciel de classification interactive MAP. Les classes en présence sont difficiles à isoler à cause des points se situant entre ces dernières qui correspondent aux pixels de transitions entre les régions homogènes.

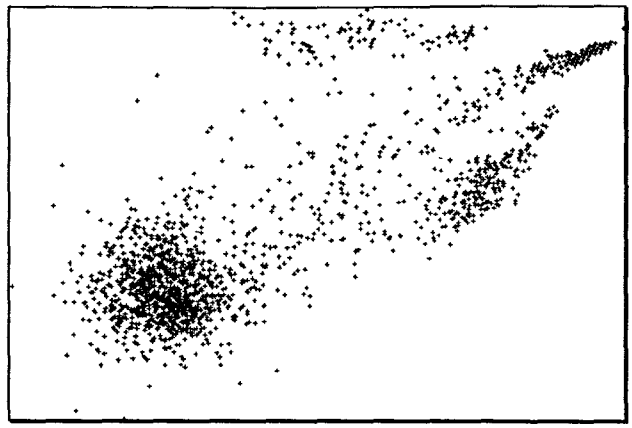


Fig. 81 : Représentation heuristique par MAP des pixels couleur dans l'espace RVB

V.2.4.2 Suppression des points de contours

Afin de supprimer les points de transitions, nous devons créer une image contour binaire qui servira de masque. L'image initiale étant multispectrale, on applique un opérateur gradient sur les trois plans rouge, vert et bleu. Soient G[R], G[V] et G[B] les trois images gradients ainsi obtenues. On construit une image gradient résultante G[R,V,B] en prenant, en chaque point, la valeur maximale des amplitudes dans les images G[R], G[V] et G[B]. Ainsi, tout contour, dans au moins un des trois plans R.V.B., est pris en compte. Pour binariser l'image gradient obtenue, nous utilisons une technique d'étiquetage probabiliste itératif [ULT94] qui permet de réduire les ambiguïtés dans les problèmes de décision en utilisant l'information contextuelle.

Après élimination des points de contours, le résultat des données fait apparaître deux classes prépondérantes et bien identifiées qui correspondent aux points blancs et aux points noirs (voir figure 82).

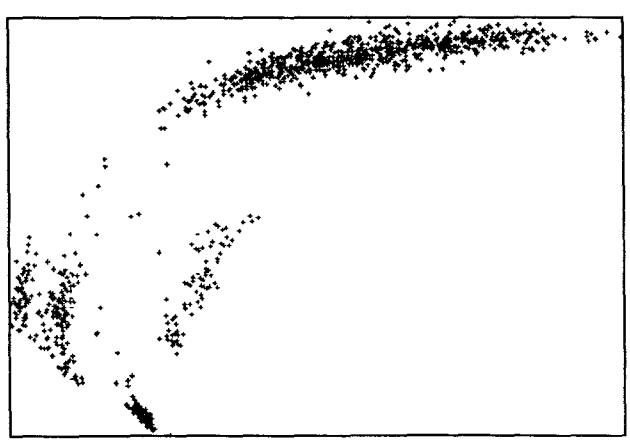


Fig. 82 : Représentation par MAP des pixels couleur après élimination des points contours

Les autres points ne peuvent être classés sans commettre d'erreurs importantes à cause de la faiblesse de la densité des populations. Ce fort déséquilibre des populations entre les points achromes et les points colorés ne permet pas une classification aisée. Les points chromatiques risquent d'être considérés comme du bruit par rapport aux deux autres classes imposantes. Pour classer les points chromatiques, il est donc nécessaire d'éliminer les points achromes.

V.2.4.3 Suppression des points achromes

Les points achromes sont divisés en deux classes : la classe des points noirs et celle des points blancs ou gris. Les points noirs sont caractérisés par une luminance faible par rapport à l'ensemble des autres points. Pour les identifier, on calcule l'histogramme du plan de luminance définie par l'équation suivante: L=(R+V+B)/3. Le premier mode de l'histogramme correspond aux points noirs. On effectue un seuillage par la méthode décrite dans [MAC93].

Le seuil est d'autant plus facile à trouver que les points de transitions ont été supprimés. Ainsi, nous pouvons éliminer l'ensemble des pixels de la classe des points noirs. Les points blancs quant à eux, sont caractérisés par une saturation faible par rapport aux points colorés.

Nous travaillons alors dans le plan de saturation, défini par l'équation: S=1-min(R,V,B)/L, sans tenir compte des points noirs, car la saturation de ces points est indéterminée. De la même manière que précédemment, le premier mode de l'histogramme correspond aux points blancs ce qui permet de séparer les points blancs des points colorés.

La figure 83 représente les différentes classes de pixels colorés qui restent après suppression des pixels achromes. La dispersion des points d'une même classe est importante, vraisemblablement due aux variations de luminance à l'intérieur d'une même couleur. L'élimination de l'effet de luminance permet de résoudre ce problème.

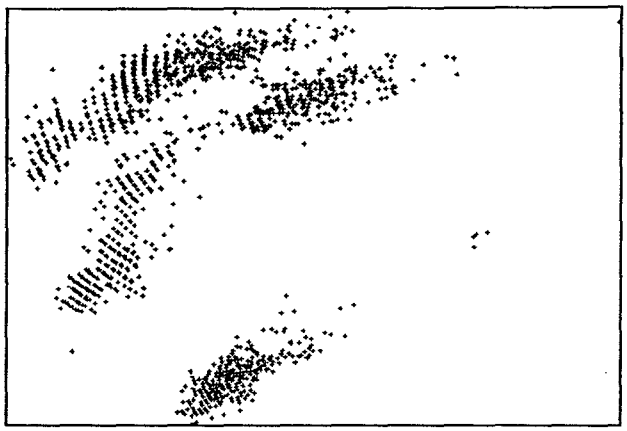


Fig. 83 : Représentation par MAP des pixels couleur après élimination des points de contour et des points achromes

V.2.4.4 Traitements des points colores

Pour supprimer l'information luminance, on utilise le système R.V.B. normalisé (r, v, b). La figure 84 montre que la classification ne pose plus de problème aussi bien au niveau des frontières interclasses qu'au niveau du nombre de classes en présence. Une méthode classique de classification peut maintenant être utilisée pour définir les différentes classes en présence.

Fig. 84 :Représentation par Map des informations colorimétriques après le dernier traitement

V.2.5 *Segmentation interactive d'images brutes*

De manière plus expérimentale, nous avons tenté d'effectuer une classification directe à partir de la représentation plane des composantes R,V,B des pixels d'une portion d'image de mosaïque traitée précédemment. De fait, nous perdons en la matière, une source d'information importante pour la segmentation qu'est l'information spatiale la figure 85 nous montre une représentation de l'ensemble des pixels dans l'espace R,V,B.

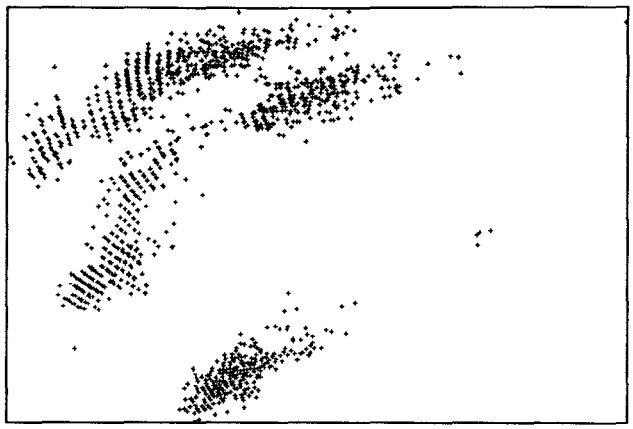


Fig. 83 : Représentation par MAP des pixels couleur après élimination des points de contour et des points achromes

V.2.4.4 Traitements des points colores

Pour supprimer l'information luminance, on utilise le système R.V.B. normalisé (r, v, b). La figure 84 montre que la classification ne pose plus de problème aussi bien au niveau des frontières interclasses qu'au niveau du nombre de classes en présence. Une méthode classique de classification peut maintenant être utilisée pour définir les différentes classes en présence.

Fig. 84 :Représentation par Map des informations colorimétriques après le dernier traitement

V.2.5 *Segmentation interactive d'images brutes*

De manière plus expérimentale, nous avons tenté d'effectuer une classification directe à partir de la représentation plane des composantes R,V,B des pixels d'une portion d'image de mosaïque traitée précédemment. De fait, nous perdons en la matière, une source d'information importante pour la segmentation qu'est l'information spatiale la figure 85 nous montre une représentation de l'ensemble des pixels dans l'espace R,V,B.

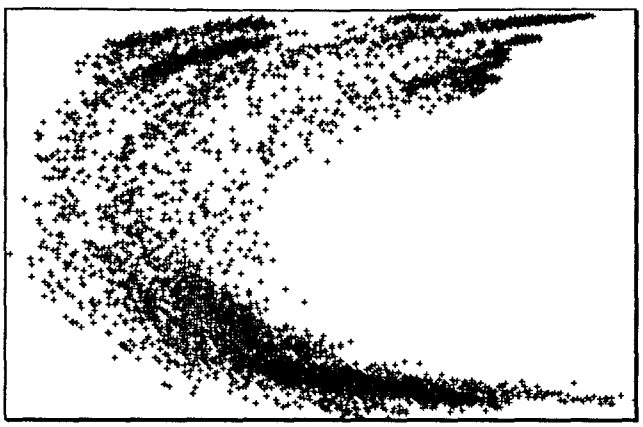


Fig. 85 : Représentation heuristique des observations colorimétriques

Les figures 86 a, b, c, d, e, f, correspondent aux classes 0,1,2,3,4,5, isolées à l'aide du logiciel de classification interactive MAP. Pour plus de clarté, celles-ci sont représentées avec la même configuration du référentiel mobile, qui est aussi celle de la figure 85.

A partir de la représentation de la figure 85, nous avons d'abord isolé la classe de points la plus importante soit la classe 0 (Fig.86a)

Puis en éliminant successivement de la représentation les classes isolées, en utilisant la démarche « coarse to fine » ainsi qu'une représentation adéquate, pour chaque nouvelle classification, nous avons isolé les 4 autres classes (Fig. 86b à 86d).

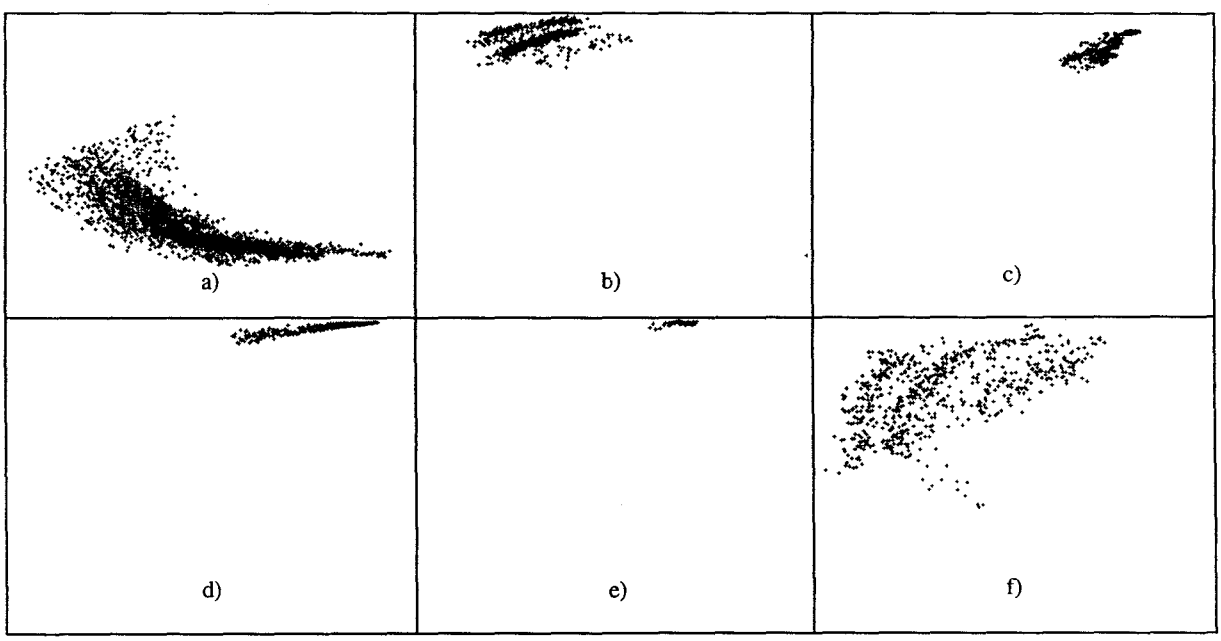


Fig. 86 : Isolation des classes d'observations

A partir de cette classification, nous avons reconstitué une image en attribuant à chaque classe de points un niveau de gris.

Les 5 classes de points sont donc représentées sur l'images en faux niveaux de gris, par des niveaux de gris croissants

On peut voir sur la figure 87 cette image en « faux niveaux de gris » (coin supérieur gauche) représentée avec les trois composantes RVB de l'image originale : R : coin supérieur droit, V : coin inférieur gauche, B : coin inférieur droit.

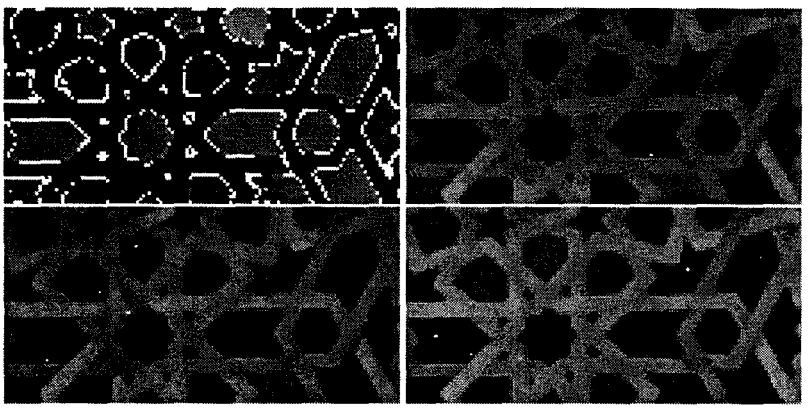


Fig. 87 : Représentation de l'image traitée en faux niveaux de gris et des trois composantes RVB

On remarque que d'une part, une classification a pu être effectuée de manière correcte sans l'aide de l'information spatiale, d'autre part, que la classe des points la plus dispersée dans la représentation, est bien la classe 5 (en blanc dans l'image en faux niveaux de gris), ce sont donc bien les contours (aux erreurs de classification près)..

V.2.6 *Conclusion*

Nous désirions utiliser une méthode de classification pour regrouper les points de même couleur d'une image de mosaïque. A l'aide de MAP, nous montrons qu'il est difficile de définir avec précision les frontières interclasses. Les traitements effectués, c'est-à-dire la suppression des points de contours et des points achromes, et la normalisation des couleurs permettent une classification aisée. Une fois la classification terminée, il faut reconstruire l'image en ajoutant les deux classes des points achromes et les points de contours non classés. Connaissant les classes, on peut calculer leurs moyennes afin d'affecter les points non classés à la classe de couleur moyenne la plus proche.

De plus, les résultats obtenus à l'aide uniquement du logiciel de classification interactive permettent d'envisager de pousser plus avant la prospective dans le domaine, en améliorant entre autre la technique de classification à l'aide de critères d'aide au choix de la représentation, ainsi que de la classification de type "coarse to fine" (Isolation de sous-classes de points de données).

VI. OUVERTURES / CONCLUSION

VI.1 Ouvertures

VI.1.*1* *Aide à la décision*

VI.1.*1*.1 Choix d'un critère adéquat

Notre logiciel actuel fait appel exclusivement à l'expertise de l'utilisateur, et à son pouvoir de discernement pour obtenir une représentation des observations multidimensionnelles lui permettant d'effectuer une bonne classification.

Il serait cependant utile d'aider ce dernier en lui fournissant une série de représentation « parlantes » qu'il pourrait éventuellement affiner.

Dans ce but nous souhaitons par la suite implanter une fonction faisant appel à des critères mathématiques d'aide à la décision.

Nous avons passé en revue de nombreux algorithmes de représentations planes. Celui qui a attiré notre attention est l'algorithme des Projections révélatrices [FTU74]. En effet, on a pu remarquer, que lorsque l'on veut classer un ensemble d'observations visuellement, on recherche des représentations le moins « normales » des données (« normales » est à prendre au sens statistique du terme) . Nous cherchons des représentations où les points écrans résultant de la transformation occupent tout l'écran, et où, dans cette distribution qui se répartit sur tout l'espace 2D, apparaissent cependant des zones de forte concentration. Les représentations 2D des observations présentent ainsi à la fois une densité locale forte et une densité globale faible.

Pour obtenir ce résultat, le critère utilisé dans l'algorithme des projections révélatrices nous semble tout particulièrement indiqué. Nous avons déjà vu ce dernier dans sa forme spécifique en (§II.3.1.4, pp. 35,36) .

Ce critère peut s'exprimer de manière plus générale et s'adapter à notre représentation sous la forme :

$$I(E) = s(E)d(E) \qquad \textbf{Equ. 70}$$

où E est l'état présent du référentiel mobile de la représentation angulaire, défini par la position de l'origine et la direction du vecteur de référence (Cf. Chap. III). Pour être plus précis, considérons $\Gamma(\alpha,\rho)$ l'image d'une observation multidimensionnelle P par notre représentation angulaire, dans une configuration E du référentiel mobile.

$\overline{\Gamma}$ est la moyenne élaguée des points images (Cf. §II.3.1.4),

soit $$\overline{\Gamma} = (\overline{\alpha},\overline{\rho}) \text{ avec. } \overline{\alpha} = \sum_{i=pn}^{(1-p)n} \alpha_i \text{ et } \overline{\rho} = \sum_{i=pn}^{(1-p)n} \rho_i \qquad \textbf{Equ. 71}$$

On rappelle que les expressions des fonctions f et h, ainsi que les détails concernant la fraction p des observations exclues des calculs, ont déjà été précisés en §II.3.1.

En reprenant la notation de Friedman et Tukey [FTU74], on peut appliquer le critère I(E) suivant soit α, soit ρ, soit les deux.

Les deux éléments du critère des projections révélatrices deviennent alors :

■ En ne se basant que sur α :

$$s_\alpha(E) = \left[\frac{\sum_{i=pn}^{(1-p)n} (\alpha_i - \overline{\alpha})^2}{(1-2p)n} \right]^{1/2}, \; d_\alpha(E) = \sum_{i=1}^{n}\sum_{j=1}^{n} f(r_{ij})h(R-r_{ij}), \text{ avec } r_{ij} = \left|\alpha_i - \alpha_j\right| \qquad \textbf{Equ. 72}$$

■ En ne se basant que sur ρ :

$$s_\rho(E) = \left[\frac{\sum_{i=pn}^{(1-p)n} (\rho_i - \overline{\rho})^2}{(1-2p)n} \right]^{1/2}, \; d_\rho(E) = \sum_{i=1}^{n}\sum_{j=1}^{n} f(r_{ij})h(R-r_{ij}), \text{ avec } r_{ij} = \left|\rho_i - \rho_j\right| \qquad \textbf{Equ. 73}$$

■ En se basant à la fois sur α et ρ:

$$s_{\alpha,\rho}(E) = s_\alpha(E)s_\rho(E), \text{ et } d_\rho(E) = \sum_{i=1}^{n}\sum_{j=1}^{n} f(r_{ij})h(R-r_{ij}) \qquad \textbf{Equ. 74}$$

$$\text{avec } r_{ij} = \left[\left(\alpha_i - \alpha_j\right)^2 + \left(\rho_i - \rho_j\right)^2 \right]^{1/2} \qquad \textbf{Equ. 75}$$

VI.1.*1*.2 Mise en œuvre du critère choisi

Du fait de la non linéarité de notre représentation et du nombre de paramètres la régissant (si la dimension de l'espace des observations est n, les paramètres à gérer seront au nombre de n(n-1)), il nous paraît inapproprié de tenter de trouver une solution locale au problème de l'optimisation du critère I(E) par une méthode analytique telle que celle du gradient, par exemple.

De fait, une méthode semble tout indiquée dans un cas comme le notre, c'est la méthode des algorithmes génétiques.

VI.1.*1*.3 Les algorithmes génétiques

VI.1.*1*.3.*a)Introduction*

Les algorithmes génétiques ont été développés par John H. Holland, à l'Université du Michigan, au début des années 70.

Un algorithme génétique, ou AG, est une méthode basée sur les principes de la sélection naturelle, pour rechercher la solution optimale à un problème complexe. C'est, en fait, une approche automatisée intelligente de la démarche heuristique (Trial and Error)[MIT95][ALA97].

Lorsque l'on se trouve, par exemple, comme souvent dans notre cas, face à un problème d'optimisation de, disons 10 paramètres, dont chacun peut prendre 100 valeurs, l'espace d'étude de la solution est particulièrement étendu, puisque le nombre possible de combinaisons est 100^{10}.

Une recherche de solutions, même locales, à ce problème, par le biais d'une approche mathématique semble exclue, à cause des temps de calcul prohibitifs, qui, de plus, augmenteraient de manière exponentielle avec le nombre de paramètres à étudier.

Un algorithme génétique, en utilisant les principes de la sélection naturelle, contourne le problème, en procédant de la manière suivante :

Tout d'abord, un ensemble de « solutions » (ou *population*) de départ est créé, en donnant aux paramètres des valeurs aléatoires appartenant à l'espace de recherche.

A partir de cette population de solutions, les plus mauvaises sont éliminées, et les meilleures sont « croisées » entre elles, en mélangeant leurs paramètres, ou « gènes » ; on crée ainsi une nouvelle population. En addition à ces « croisements », de temps à autre, un « gène » est modifié légèrement, pour produire une « mutation ». Comme dans les processus de la vie, ce type d'adaptation crée un « organisme » très robuste. L'ensemble du processus se prolonge sur plusieurs « générations », et les meilleurs gènes sont transmis de l'une à l'autre.

Le résultat est, dans la plupart des cas, une très bonne solution au problème. Ces algorithmes permettent de résoudre des problèmes considérés auparavant comme trop compliqués ou trop vastes. De plus, ces algorithmes sont très utiles dans le domaine particulièrement complexe des problèmes non linéaires auquel apparient notre algorithme.

VI.1.*1*.3.*b)Terminologie*

•Comparons la terminologie de la biologie avec celle des AG

Biologie	**Algorithme Génétique**
chromosome	chaîne
gène	paramètre, caractéristique ou détecteur
allèle	valeur de paramètre
locus	position dans la chaîne
génotype	structure
phénotype	ensemble de paramètres, solution alternative, structure décodée
épistasie	non linéarité

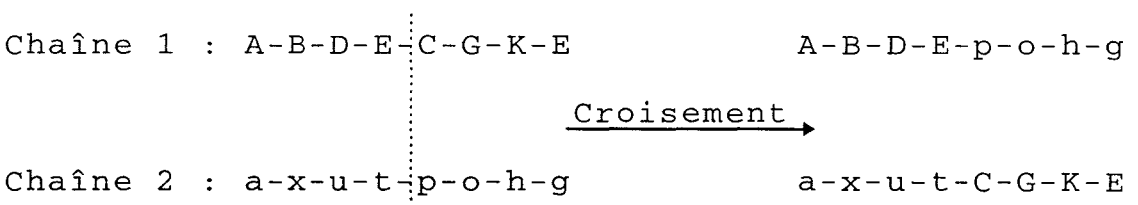
•Définition des différents termes

***Allèle :**

Les différentes formes d'un gène sont connues sous le nom d'allèles. Les allèles produisent des différences dans l'ensemble des caractéristiques associées au gène.

***Croisement**

Le croisement débute avec la sélection aléatoire d'un entier dont la valeur est inférieure à la longueur d'une chaîne. Deux chaînes sont accouplées en joignant le préfixe de l'une avec le suffixe de l'autre, par rapport au point de croisement. Donnons un exemple :



***Gènes**

Dans la terminologie biologique, on dit que les gènes sont formés de **chromosomes** qui peuvent prendre un certain nombre de valeurs appelées **allèles**. Dans le cas des algorithmes génétiques, on dira que les chaînes sont composées de **paramètres** ou **détecteurs** qui peuvent prendre différentes valeurs.

***Génotype**

la structure génétique est appelée **génotype**.

***Mutation**

La mutation est employée pour éviter l'écueil d'une solution convergeant vers un minimum local quand la recherche se fait sur un espace peu dense avec un nombre important de paramètres.

On attribue généralement à une mutation une faible probabilité d'occurrence, dans l'exemple qui suit, elle sera, par exemple, de l'ordre de 0.0001 par bit.

Exemple de mutation:

Chaîne avant mutation : 011011001

Chaîne après mutation 011011011

L'avant-dernier bit de la chaîne a *muté* de 0 à 1.

***Phénotype**

Une instanciation particulière de **génotype** est un **phènotype**.

***Reproduction**

Dans l'étape de reproduction du cycle d'évolution d'une génération, on attribue à chaque chaîne individuelle une valeur proportionnelle à sa proximité avec la solution : une valeur d'*aptitude* ou encore « fitness ». La probabilité de reproduction est proportionnelle à cette valeur.

Exemple de reproduction:

#	Chaînes de la première génération	Fitness		Chaînes de la seconde génération	#
1	0 1 1 1 0 1 0 1	61		1 1 1 0 0 1 0 1	4
2	0 1 1 0 1 0 0 1	34		0 1 1 0 1 0 0 1	2
3	0 1 0 1 0 1 1 0	27		0 1 1 1 0 1 0 1	1
4	1 1 1 0 0 1 0 1	30	Reproduction →	1 1 1 0 1 1 1 1	5
5	1 1 1 0 1 1 1 1	41		0 1 1 1 0 1 0 1	1
6	1 0 1 1 0 1 0 1	17		1 1 1 0 1 1 1 1	5
7	1 1 0 1 1 1 0 0	9		0 1 1 1 0 1 0 1	1

On remarque que la chaîne 1 e la première génération se trouve reproduite trois fois dans la deuxième, la chaîne 5 deux fois.. etc... Les chaînes 3, 6, et 7 par contre, ont disparu, puisque leur « fitness » était trop faible.

VI.1.2 *Extensions Possibles des fonctionnalités de l'interface*

Les extensions des fonctionnalités de notre logiciel de représentation et de classification interactives MAP sont très nombreuses. La programmation sous l'interface Windows nous permet effectivement de nombreuses ouvertures : Les améliorations sont de deux ordres :

- Les améliorations de type théorique, ce sont les critères d'aide à la décision, exposés au paragraphe précédent,
- Les améliorations de type pratique, qui nous paraissent toutes aussi urgentes, car ce sont elles qui permettront à MAP de s'intégrer pleinement au sein des autre logiciels professionnels Windows :

☞ **Le module d'impression** n'a pas encore été implanté. Il est nécessaire pour que cette application rentre pleinement dans le cadre des applications graphiques Windows

☞ **L'échange de donnée DDE**[35] n'est pas encore géré, or il serait intéressant de le mettre en place pour permettre à notre application de communiquer « en temps réel » avec des applications comme les tableurs, les bases de données, ou simplement les applications graphiques (Mise à jour des données dans une application lorsque celles-ci sont modifiées dans une autre, « tiré-déposé » (« drag and drop »)... etc.

☞ **Les filtres d'importation** sont aussi une possibilité, pour permettre à Map de représenter des données en provenance, là aussi, de tableurs, logiciels de bases de données, ou de logiciels de traitement de texte (Excel, Dbase, Word)

☞ **L'interface** devra à terme évoluer vers une interface **MDI**[36], comme quasi toutes les applications modernes implantées en 32 bits sous Windows95 ou WindowsNT. Il sera alors possible d'étudier et de classer plusieurs instances du même fichier de données, et de les comparer, ou tout simplement de travailler sur plusieurs fichiers à la fois.

---

[35] Direct Data Exchange
[36] Multiple Document Interface

Map évoluera aussi vers une interface orientée « document » et non « application » , forme qu'elle a héritée de sa conception première sous Windows3.1.

VI.2 Conclusion

Lorsque nous avons conçu notre mode de représentation angulaire, ainsi que le logiciel interactif MAP qui permet d'exploiter celui-ci, notre but était de fournir à l'utilisateur, non spécialiste dans le domaine de l'analyse de données, un outil efficace d'inspection et de classification des données multidimensionnelles quantitatives en sa possession. Nous voulions aussi fournir aux chercheurs en analyse de données un outil rapide et fiable de représentation des observations qui leur permette d'effectuer aisément une vérification croisée avec leurs propres algorithmes.

Notre but était donc multiple :

- efficacité,
- simplicité,
- ergonomie,
- portabilité,

En mettant au point MAP, nous nous sommes beaucoup rapprochés de ces divers objectifs.

Il reste cependant, comme nous l'avons vu dans les deux paragraphes précédents, des améliorations a apporter à notre travail, pour permettre à ce logiciel original de s'inscrire pleinement dans la gamme des outils reconnus dans le domaine de l'analyse de données.

VII. BIBLIOGRAPHIE

1. [**AKA74**] H. Akaike, "A New look at the Statistical model Identification", IEEE Tans. Appl. Comput., Vol. AC-19, pp. 716-723, 1974
2. [**ALA97**] J. T. Alander, "An Indexed Bibliography of Genetics Algorithms and Optimization", Report Series N° 94-1-OPTIMI, Dept. Of Information Technology and Production Economics, University of Vaasa, Finland, April 1997.
3. [**AND73**] M.R. Andenberg, "Cluster Analysis for Applications", Academic Press Inc., New York, 1973.
4. [**ASS89**] J.P. Asselin de Beauville, "Panorama sur l'utilisation du mode en classification automatique", RAIRO-APII, AFCET, n°2, pp. 113-137, 1989.
5. [**BAH89**] P. Baldi and K. Hornik, "Neural Networks and Principal Component Analysis : learning from examples without local minima", Neural Networks, Vol. 2, n°1, pp. 53-58, 1989.
6. [**BAH91**] P. Baldi and K. Hornik, "Back-propagation and Unsupervised Learning in Linear Networks", Technical report, Jet Propulsion Laboratory and division of Biology, Cal. Inst. of Technology, 1991.
7. [**BAL65**] G. M. Ball, "Data Analysis in the social sciences : what about the details?", Proc. F.J.C.C. pp. 533-560,1965.
8. [**BAL67**] G.H. Ball and D. J. Hall., "A clustering technique for summarising multivariate data", Behavioural Science, Vol. 12, pp. 153-155, 1967.
9. [**BAN88**] H. Bandemer and W. Näther, "Fuzzy Projection Pursuits", Fuzzy Sets and Systems, Vol. 27, pp. 141-147, North Holland, 1988.
10. [**BAY80**] C. H. Bayne, J.J. Beauchamp, C. L. Begovitch and V.E. Kane, "Monte Carlo comparison of selected clustering procedures", Pattern Recognition, Vol. 12, pp. 51-62, 1980.
11. [**BDH95**] M. Bétrouni, S. Delsert and D. Hamad, "Interactive Pattern Classification by Means of Artificial Neural Networks", 1995 IEEE. Int. Conf. on Syst., Man and Cyb. Proc., Vol. 4, pp. 3275-3279, 1995.
12. [**BJD81**] G. Biswas, A. K. Jain and R. C. Dubes "Evaluation of projection algorithms", IEEE. Trans. Pattern Anal. and Mach. Intell., PAMI-3, pp. 701-708, 1981.
13. [**BOC79**] H. H. Bock, "Clustering by Density Estimation", Analyse de Donnée et Informatique, INRIA, pp. 173-186, 1979.
14. [**BOT91**] C. Botte Lecocq, "L'analyse de données multidimensionnelle par transformation morphologique binaire", Thèse de Doctorat, Université des Sciences et Technologies de Lille, 1991.
15. [**BOU88**] H. Boulard and Y. Camp, "Auto-association by the Multilayer Perceptrons and Singular Value Decomposition", Biological Cybernetics, Vol. 59, pp. 291-294, 1988.
16. [**BUR09**] C. Burt, "Experimental tests of general intelligence," Brit. Jour. Psychol. Vol.3 pp. 94-177, 1909.
17. [**CMN83**] S. Card, T. Moran and A. Newell, D. B. Cooper and P. W. Cooper, " The Psychology of Human-Computer Interaction" , Chapter 2, " The Human Information Processor", pp. 23-97, Erlbaum, 1983.
18. [**COO64**] D. B. Cooper and P. W. Cooper, "Nonsupervised adaptive signal detection and pattern recognition", Information and Control, Vol. 7, pp. 416-444,1964.
19. [**COO67**] P. W. Cooper, "Some topics on nonsupervised adaptative signal detection for multivariate normal distributions", Computer and Information Sciences, Vol. II, pp. 123-146, Academic Press, New-York, 1967.

20. [**CRA91**] M. A. Cramer, "Non Linear Principal Component Analysis Using auto-associative Neural networks", AICHE Journal, Vol. 37, n°2, pp. 233-243, February 1991.

21. [**DAA93**] M. Daoudi, D. Hamad and J-G. Postaire, "Interactive Classification through Neural Networks", Artificial Neural Nets and Genetic Algorithms, Proceedings of the International Conference, pp. 80-85, R.F. Albrecht, C.R. Reeves and N.C. Steele (eds.), Innsbruck, Austria, 1993.

22. [**DAB93**] M. Daoudi, R. Benslimane, D. Hamad and J-G. Postaire, "A New Interactive Pattern Recognition Approach by Multilayer Neural Networks and Mathematical Morphology", IEEE. Int. Conf. on Syst., Man and Cyb. Proc., Vol. 4, pp. 125-130.

23. [**DAC93**] E. Diday, "A Display Oriented Technique for Interactive Pattern Recognition by Multilayer Network", IEEE Int. Conf. on Neural Networks, San Francisco, California , Vol. III, pp. 1633-1637, 1993.

24. [**DAL62**] R.F. Daly, "The Adaptive Binary - detection problem on the real line", Technical Report 2003-3, Stanford University, Stanford, California, 1962.

25. [**DAV88**] M. L. Davison, "Multidimensional scaling", Wiley, New York,1983.

26. [**DAY69**] N. E. Day, "Estimating the components of a mixture of normal distributions", Biometrika, Vol. 56, pp. 463-474,1969.

27. [**DID71**] E. Diday, "Une nouvelle méthode en classification automatique et reconnaissance des formes: La méthode des nuées dynamiques", Rev. Stat. Appl., Vol. 19; pp. 29-33, 1971.

28. [**DID82**] E. Diday, J. Leman, J. Pouget et F. Testu, "Eléments d'analyse de données", Bordas, Paris, 1982.

29. [**DRO8X**] J. Droesbeke, B. Fichet et P. Tassi, "Modèles pour l'analyse des données multidimensionnelles", Economica, Paris, 198X.

30. [**DUD73**] R. O. Duda and P. E. Hart, "Pattern classification and scene analysis", Wiley, New-York 1973.

31. [**ESS92**] F. Esson and J.G. Postaire "An Interactive Tool For The Classification of Multivariate Data through a Virtual Observer." ITI'92 Proc. pp. 319-323, 1992

32. [**ESS93**] F. Esson and J.G. Postaire, "A New Interactive Tool For The Classification of Multivariate Data through a Virtual Observer.", IEEE SMC93 Proc. pp 109-113

33. [**UTE95**] V. Utré, F. Esson, J.G. Postaire, "Prétraitement des Images Couleurs en Vue de la Segmentation par Classification des pixels. Application à l'Analyse d'Images des mosaïques." Conférence MCEA-95, Grenoble, les 13-15 Septembre 1995.

34. [**UFE95**] V. Utré, F. Esson, J.G. Postaire, "Segmentation d'Images Couleurs par Classification Interactive". Colloque PR-SEA, Bourges, les 1er et 2 Juin 1995.

35. **EVE77**] Brian Everitt, "Cluster Analysis", Halsted Heinemann, 1977.

36. [**FEH78**] J. Fehlauer and B. A. Eisenstein, "A declustering criterion for feature extraction in pattern recognition", I.E.E.E. Trans. Computers, Vol. C-27, pp. 261-266, 1978.

37. [**FFT74**] M. A. Fisherkeller, J.H. Friedman and J.W. Tukey, "PRIM-9 : an Interactive Multidimensional Data Display and Analysis System," A.E.C. Scientific Computer Information Exchange Meeting, May 2-3 1974 / The collected works of John W. Tukey(1988) Wadsworth, Inc. Belmont USA, Sept. 1974.

38. [**FIS36**] R. A. Fisher, "The use of multiple measurements in taxonomic problems", Ann. Eugenics, Vol. 7, pp. 178-188, 1936.

39. [**FIZ87**] H. Fizazi, "Classification automatique de petits échantillons de grande dimension. Application à la biométrie de l'Abeille" Thèse de Doctorat, Université des Sciences et Techniques de Lille, 1987.

40. [**FRE81**] J. Fresnay, "Biométrie de l'abeille" ,2^{éme} édition, INRA, OPIDA, 1981.

41. [**FRI67**] H.P. Friedman and J. Rubi, "On some invariant criteria for grouping data", J. American Statistical Assn., Vol. 62, pp. 1059-1118, 1967.

42. [**FRI87**] J.H. Friedman, "Exploratory Projection Pursuit", J. American Statistical Assn., Vol. 82, pp. 249-266, 1987.

43. [**FRO74**] F.R. From and R. A. Northouse, "Class: a non parametric clustering algorithm", Pattern Recognition, Vol. 8, pp. 107-114, 1974.

44. [**FRR87**] H.P. Friedman an J. Rubi, "On some invariant criteria for grouping data", J. American Statistical Assn., Vol. 62, pp. 1059-1118, 1967.

45. [**FTU74**] J.H. Friedman and J.W. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis", IEEE Trans. Comput., Vol. C-23, pp. 881-889, Sept 1974.

46. [**FUK75a**] K. Fukunaga and L.D. Hostetler, "The estimation of the gradient of a density function with applications in pattern recognition", IEEE Trans. Info. Theory, Vol. IT-21, n°1, pp. 32-40, 1975.

47. [**FUL70**] K. Fukunaga and D. R. Olsen, "A Two Dimensional Display for the Classification of Multivariate Data", IEEE Trans. Comput., Vol. C-20, pp. 917-923, Dec. 1970.

48. [**FUM82**] K. Fukunaga and J. M. Mantock, "A Nonparametric Two-Dimensional Display for Classification", IEEE Trans. Patt. Anal. & Mach. Intell., Vol. PAMI-4, n°4, pp. 427-436, July 1982.

49. [**FUM84**] K. Fukunaga and J. M. Mantock, "Nonparametric Data Reduction", IEEE Trans. Patt. Anal. & Mach. Intell., Vol. PAMI-6, n°1, pp. 115-118, 1984.

50. [**GEL80**] E. Gelsema and R. Eden, "Mapping algorithms in ISPAHAN", Pattern Recognition, Vol. 12, pp. 127-136, 1980.

51. [**HAS66**] V. Hasselbald, "Estimation of parameters for a mixture of normal distributions", Technometrics, Vol. 8, pp. 431-444, 1966.

52. [**HBE95**] D. Hamad and M. Bétrouni, "Artificial Neural Networks for Nonlinear projection and Exploratory Data", Alès, 1995.

53. [**HDE95**] D. Hamad and S. Delsert, "Nonlinear mapping procedures for data analysis", Int. Conf. on Engineering Applications of Neural Networks (EANN '95) proceedings, Otaniemi, Espoo, Finland, pp. 457-460, 1995.

54. [**HDE96**] D. Hamad and S. Delsert, "Exploratory Data Analysis by Means of Artificial neural Networks", IEEE-SMC CESA IMACS Multiconference, Lille France, pp. 649-653, 1996.

55. [**HIL68**] C. G. Hillborn and D.G. Lainiotis, "Optimal Unsupervised Learning Multicategory Dependant Hypotheses Pattern Recognition", IEEE Trans. Info. Theory, Vol. IT-14, pp. 468-470, 1968.

56. [**HUB85**] P. J. Huber, "Projection pursuit", The Annals of Stat., Vol. 13, pp. 435-475, 1985.

57. [**HUG80**] R. A. Hughes, "Analysis of semiconductor test data using pattern recognition techniques", INSPEC Conference paper Issue 804 81053496 Sponsor: IEEE, 1980.

58. [**JAI88**] A. K. Jain and R.C. Dubes, "Algorithms for Clustering Data", Prentice Hall, pp. 130-133, 1988.

59. [**JAM92**] A.K. Jain and J. Mao, "Artificial Neural Networks for Non-linear Projection of Multivariate Data", IJCNN, Vol. III, pp. 335-340, Baltimore, Maryland, 1992.

60. [**JMG95**] J.M. Gillet, "L'interface graphique : enjeux, ergonomie, mise en œuvre sous Windows", InterEditions, 1995.

61. [**JON68**] K.L. Jones, "Problems of grouping individuals and the method of modality", Behavioral Science, Vol. 13, pp. 496-511, 1968.

62. [**JOS87**] K.L. Jones and M.C. Sibson, "What is Projection Pursuit", J. Roy. Stat. Soc. ser. A., pp. 1-38, 1987.

63. [KER78] B. W. Kerninghan and D. M. Ritchie, "The C Programming Language", Prentice-Hall, 1988.

64. [KHF86] D. K. Kahaner, J. Horlick and D. K. Foer, "Mathematical Software in Basic : RV, Generation of Uniform and Normal Random Variables", IEEE MICRO, June 1986.

65. [KIT76] J. Kittler, "A Locally Sensitive Method for Cluster Analysis", Pattern Recognition , Vol. 8, pp. 23-33, 1976.

66. [KOH90] T. Kohonen, "The Self Organizing Map", IEEE Proc., Vol. 78, n°9, pp. 1464-1480, 1990.

67. [KOO76] W.L.G. Koonitz, P.M. Narendara and K. Fukunaga, "A graph theoretic approach to non parametric cluster analyser", I.E.E.E. Trans. Comput., Vol. C-25, pp. 936-944, 1976.

68. [LAW67] G.N. Lane and W.T. Williams, "A general theory of classificatory sorting strategies 1.Hierarchical systems", Computer J., Vol. 9, pp. 973-980, 1967.

69. [LEC86] Y. Le Cun, "Learning process in asymmetric threshold network", Disorder Systems and Biological Organisation, E. Bienenstock, F. Fogelman Solute and G. Weibush (Eds.), Berlin : Springer, 1986.

70. [LIP87] R. P. Lippman, "An introduction to computing with neural nets", IEEE ASSP Magazine, pp. 4-222, 1987

71. [LOF65] D.O. Loftsgaarden and C. P. Quesenberny, "A Non Parametric Estimate of a Multivariate Density Function", Ann. Math. Stat., Vol. 36, pp. 1049-1051, 1965.

72. [LSB76] R. C. T. Lee, J. R. Slagle, and H. Blum, "A Triangulation Method for the Sequential Mapping of Points from N-Space to Two Space", IEEE Trans. Comput., Vol. C-26 pp. 288-292, 1976.

73. [LSB79] R. C. T. Lee, J. R. Slage and H. Blum, "A triangulation method for the sequential mapping of points from N-space to two-space," IEEE Trans. Comput., Vol. C-26, pp. 288-292, 1979.

74. [LUK79] A. Lukasova, "Hierarchical agglomerative clustering procedure", Pattern recognition, Vol. 11, pp. 365-381, 1979.

75. [MAC67] J. Macqueen, "Some methods for the classification and analysis of multivariate observations", Proc. 5$^{Th.}$ Berkeley Symp. on Math. Stat. and Prob., University of California Press, Berkeley and Los Angeles, Vol. 1, pp. 281-297, 1967.

76. [MAK77] U. E. Makov and A.F.M. Smith, "A quasi Unsupervised Learning Procedure for Priors", IEEE Trans. Info. Theory, Vol. IT-24, n°6, pp. 761-764, 1977.

77. [MCD83] J. A. Mc Donald, "Orion I : Interactive Computer Graphics in Statistics", Naval Research Review (USA), Vol. 35, n°2, pp. 29-32, 1983.

78. [MIT95] M. Mitchell, "An Introduction to Genetic Algorithms", Complex Adaptive Series, A. Bradford Books, December 1995.

79. [MIX82] D. F. Mix and R. A. Jones, "A dimensionality reduction technique based on a least squared error criterion", IEEE Trans. Patt. Anal. & Mach. Intell., Vol. PAMI-4, pp. 537-544, 1982.

80. [MIZ75] R. Mizoguchi and M. Shiruma, "An Approach to Unsupervised Learning Classification", IEEE Trans. Comput., Vol. C-24, n°10, pp. 979-983,1975.

81. [NIW79] H. Niemann and J. Weiss, "A fast converging algorithm for nonlinear mapping of high-dimensional data to a plane", IEEE Trans. Comput., Vol. C-28 pp. 142-147, 1979.

82. [OLE88] S. Olejnik, "Analyse de la convexité d'une fonction de densité de probabilité par étiquetage probabiliste : application à la classification automatique non supervisée", Thèse de Doctorat, Université des Sciences et Techniques de Lille, 1988.

83. [PAR62] E. Parzen, "On Estimation of a Probability Density Function and Mode", Am. Math. Stat., Vol. 33, pp. 1065-1076, 1962.

84. [**PO187**] J. G. Postaire, "Optimisation du processus de classification automatique par analyse de la convexité des fonctions de densité de probabilité", Thèse d'état, Université des Sciences et techniques de Lille, 1987.

85. [**PO287**] J. G. Postaire, "De l'image à la décision", Dunod Informatique, Paris 1987.

86. [**POS81**] J. G. Postaire and C. Vasseur, "An Approximate Solution to Normal Mixture Identification with application to Unsupervised Pattern Classification", IEEE Trans. Patt. Anal. & Machine Intell., Vol. PAMI-3, n°2, pp. 163-179, 1981.

87. [**POS82b**] J. G. Postaire and C. P. A. Vasseur, "A Fast Algorithm for Non Parametric Probability Density", IEEE Trans. Patt. Anal. & Machine Intell., Vol. PAMI-4, n°6, pp. 663-666, 1982.

88. [**POS89**] J. G. Postaire and A. Touzani, "Mode boundary detection by relaxation for cluster analysis", Pattern Recognition Letters, Vol. 22, pp. 477-490, 1989.

89. [**POS93**] J.G. Postaire, R. D. Zhang, and C. Botte-Lecocq, "Cluster Analysis by Binary morphology", IEEE Trans. Patt. Anal. & Machine Intelligence, Vol. PAMI-15, n°2, pp. 170-180, 1993.

90. [**POV82**] J. G. Postaire and C. Vasseur, "A Fast Algorithm for Nonparametric Probability Density Estimation", IEEE Trans. Patt. Anal. & Mach. Intell., Vol. PAMI-4, n°6, pp. 663-666, 1982.

91. [**RHW86**] D. E. Rumelhart, G. E. Hinton and R. J . Williams, "Learning internal representation by error propagation Parallel Distributed Processing", Explorations in the micro structures of cognition, MIT Press, Cambridge Mass., Vol. 1, pp. 318-362, 1986.

92. [**SAM69**] J.W. Sammon Jr., "A nonlinear mapping for data structure analysis", IEEE Trans. Comput., Vol. C-18, pp. 401-409, 1969.

93. [**SAM70b**] J.W. Sammon Jr., "An optimal discriminant plane", IEEE Trans. Comput., Vol. C-19, pp. 826-829, 1970.

94. [**SCH76**] A. Schroeder, "Analyse d'un mélange de distributions de probabilité de même type" Rev. Stat. Appl., Vol. 24, n°1, pp. 39-62, 1976.

95. [**SHE79**] R. N. Shepard and P. Arabie, "Additive Clustering : representation of similarities as combinations of discrete overlapping properties", Psychological review, n° 86, pp. 87-123, 1979.

96. [**SIE88**] W. Siedlecki, K. Siedlecka and J. Slansky, "Mapping techniques for exploratory pattern analysis", Pattern Recognition and Artificial Intelligence, pp. 277-298, 1988.

97. [**SNE73**] P.H.A. Sneath and R. R. Sokal, "Numerical Taxonomy", W.H. Freeman & Co., Publishers, San Francisco, 1973.

98. [**SUG92**] D. Sudhanva and K. Chidananda Gowda, "Dimensionality reduction using geometric projections : a new technique", Pattern Recognition, Vol. 25, n° 8, pp. 809-817, 1992.

99. [**TOU88**] A. Touzani and J. G. Postaire, "Mode Detection by Relaxation", IEEE Trans. Patt. Anal. & Machine Intell., Vol. PAMI-10, pp. 970-978, 1988.

100. [**TOU89**] A. Touzani and J. G. Postaire, "Clustering by mode boundary detection", Pattern Recognition Letters, Vol. 9, pp. 1-12, 1989.

101. [**UTR96**] V. Utré, "Contribution à la segmentation d'images de mosaïques en couleur ", Thèse de Doctorat, Université des Sciences et Technologies de Lille, 25 Janvier 1996.

102. [**VAS80**] C. Vasseur and J. G. Postaire, "A Convexity Testing Method for Cluster Analysis", IEEE Trans. Syst. Man. and Cybern., Vol. SCM-10, n°3, pp. 145-149, 1980.

103. [**WAL67**] S. Watanabe and al., "Evaluation and selection of variables in pattern recognition", Computer and Information Science, Academic Press, New York, Vol. II, pp. 91-122, J.T. Tou, Ed., 1967.

104. [**WAS89**] P. D. Wasserman, "Neural computing Theory and Practise", VNR, New York NY, 1989.

105.[**WOL70**] J. H. Wolfe, "Pattern Clustering by Multivariate Mixture Analysis", Multi. Behav. Res., Vol. 5, pp. 329-350, 1970.

106.[**WSA80**] S. K. Wismath, H. P. Soong and S. G. Akl, "Feature Selection By Interactive Clustering", Pattern Recognition, Vol. 14, pp. 75-80, 1980.