

Les techniques de visualisation de données

De manière à introduire les notions nécessaires au chapitre 6 concernant la visualisation de dissimilarité pour des MMC, nous décrivons dans ce chapitre en quoi consiste la visualisation d'information.

3.1 Introduction

Depuis des siècles, l'être humain cherche à comprendre le monde qui l'entoure. Cette compréhension se traduit principalement par la définition de règles et de lois nous permettant de mieux comprendre notre environnement, de mieux prédire l'avenir et de pouvoir confirmer les hypothèses établies sur ce monde.

Dans le cas de la fouille de données, l'être humain ne cherche plus à comprendre un environnement global, mais plutôt à comprendre une partie de cet environnement. A cet effet, il est à la recherche de structures, de caractéristiques, de motifs, de tendances, d'anomalies ou, de manière plus générale, de la présence ou de l'absence de relations dans les données (Fayyad et al., 2001).

Depuis plus d'un demi-siècle, cette analyse manuelle des données, effectuée dans le but d'en extraire des connaissances, n'est plus réalisable à la main. En effet, du fait de la multiplication des paramètres pris en compte et de la multiplication des mesures effectuées, la quantité d'information à traiter a rendu obligatoire l'utilisation de méthodes pour les analyser.

Le traitement purement automatique de ces informations afin d'en extraire des connaissances se nomme la « fouille de données » ou *data mining*. Bien que permettant une automatisation de l'analyse, la fouille de données n'est pas la solution parfaite au problème de la compréhension des données. En effet, très souvent, les algorithmes de fouilles de données, aussi sophistiqués qu'ils soient, ne parviennent pas à extraire la totalité des informations pertinentes. Cette limite peut se matérialiser, par exemple, par la définition de règles de décision non minimales (par exemple, elle est parfois atteinte par l'algorithme C4.5 (Quinlan, 1993)). De plus, l'extraction automatisée des connaissances, aussi efficace soit-elle, ne permet pas toujours de transmettre simplement ces connaissances à l'expert.

Pour résoudre ce problème, une solution viable consiste à présenter graphiquement les données de manière à permettre à la fois de guider, par l'interaction, une extraction de connaissances (automatique ou manuelle) et de permettre la transmission des connaissances à l'expert en l'aidant à construire sa carte mentale des données. Ainsi, on parlera de fouille visuelle de données ou de *visual data mining*. Le rôle de la visualisation dans la fouille visuelle de données peut se résumer en plusieurs points principaux (Fayyad et al., 2001). Elle permet d'afficher une grande quantité de données, fournissant de ce fait une vue d'ensemble des données. Elle

permet automatiquement, ou par interaction, de résumer et annoter des données. La visualisation permet également de détecter des sous-ensembles de données où il est intéressant d'appliquer tel ou tel outil de fouille de données ou de fouille visuelle de données de manière plus précise. D'un autre côté, l'expert cherchant à analyser les données peut ne pas savoir ce qu'il cherche. Dans ce cas, le choix d'un algorithme de fouille de données est difficile à faire. La visualisation trouve alors son intérêt grâce à l'interaction, permettant à l'utilisateur d'établir une carte mentale des données et de rechercher des structures.

Que ce soit pour la confirmation d'hypothèses, pour la recherche de structures ou pour l'application d'algorithmes prenant en compte l'utilisateur, l'interaction de l'utilisateur avec la visualisation tient un rôle très important. L'interaction permet, entre autres, la navigation dans les données (vue sous différents angles), l'interrogation des données (réduction de l'affichage ou de caractéristiques d'affichage à une portion de l'espace) et l'association de données entre elles (marquages, simplification, réduction, ...) (Fayyad et al., 2001).

3.2 Pourquoi la visualisation de données a-t-elle un rôle si important ?

« The history of visualization is that of the search for new artefacts to amplify the ability to know ; it's the history of writing and of maps, the history of knowledge. »
(Dursteler, 2003)

Depuis des siècles, l'être humain utilise les représentations afin d'accroître sa capacité à intégrer des informations (Dursteler, 2003). Ces représentations prennent des formes très variées : écriture (chiffre, lettre, ...), cartes, graphiques, ... On peut alors se demander : « est-ce que l'homme a vraiment besoin de ces représentations ? » La réponse est très probablement « oui ».

Historiquement, le rôle des visualisations était de permettre la découverte et la compréhension de structures de hauts niveaux dans les données, impossibles à découvrir par une analyse directe des données (Eick, 2000). Prenons l'exemple d'une fonction f dont on connaît un certain nombre de valeurs. L'analyse de ces valeurs numériques fournit peu d'information sur la fonction f . Si, au lieu d'analyser ces valeurs, on trace le graphique correspondant aux valeurs de la fonction, il devient beaucoup plus facile de comprendre le comportement de la fonction. En effet, même en présence de bruit dans les valeurs, dû par exemple à des erreurs de calcul ou de précision de calcul, la représentation permet d'obtenir visuellement le comportement global de f . Par conséquent, la visualisation d'information permet de faciliter l'abstraction des informations dans les données.

Depuis quelques années, la visualisation d'information ne sert plus uniquement à la découverte d'information dans les données. Elle sert également de moyen d'accès aux détails des données et à l'affinement des caractéristiques statistiques extraites des données (Eick, 2000). Par conséquent, il devient de plus en plus courant de rencontrer des visualisations qui permettent de confirmer, ou d'expliquer des hypothèses définies *a priori*.

3.3 Définitions et structure générale d'un système de visualisation d'information

Pour décrire la structure générale d'un système de visualisation d'information, il est nécessaire d'introduire au préalable quelques définitions.

3.3.1 Définitions

Définition 8 *Une visualisation, c'est l'ensemble des moyens graphiques autres que ceux textuels ou ceux verbaux utilisés dans le but de communiquer des informations.*

Définition 9 *L'interaction regroupe l'ensemble des moyens mis à disposition de l'utilisateur et l'ensemble des actions de l'utilisateur permettant d'ajuster une visualisation de manière interactive.*

Définition 10 *Un attribut graphique est un aspect élémentaire d'une visualisation contrôlable par l'utilisateur définissant les caractéristiques de la représentation (position, taille, couleur, forme, orientation, vitesse, texture, transparence, ...).*

Définition 11 *La mise en correspondance ou mapping est l'opération qui consiste à définir la façon dont les attributs des données sont associés aux attributs graphiques .*

Définition 12 *La normalisation des données est l'opération qui consiste à ré-échelonner les valeurs prises par les différents attributs des données dans un intervalle donné.*

Il est possible de définir principalement trois types de normalisation (Fayyad et al., 2001) :

- la normalisation locale : chaque dimension est ramenée dans l'intervalle $[0, 1]$ ou $[-1, 1]$. Ce type de normalisation confère un poids égal à chacune des dimensions ;
- la normalisation globale : toutes les dimensions sont ré-échelonnées dans l'intervalle $[m, M]$ en désignant par m le minimum (respectivement M le maximum) des valeurs sur toutes les dimensions. Ce type de normalisation donne un poids plus important à la dimension qui a la plus grande valeur ;
- la normalisation pondérée : seules les dimensions qui ont un sens équivalent sont normalisées simultanément (ex : des kilogrammes de plomb et des kilogrammes de fer, ...).

De manière générale, la normalisation locale est la plus utilisée, car la plus neutre. Cependant, dans certains cas, les autres normalisations peuvent être plus efficaces pour la visualisation.

3.3.2 Structure générale d'un système de visualisation d'information

Le processus de visualisation d'information est un processus en cinq étapes (cf. figure 3.1) interagissant entre elles (Mroz et al., 1998) :

- les deux premières étapes sont l'acquisition et la préparation des informations. Après l'acquisition des données ayant pour origines des expérimentations, des observations, voire des expérimentations numériques ou des simulations, il est nécessaire de transformer ces données dans une forme compréhensible par la méthode de visualisation. Pendant la transformation, les données peuvent être filtrées, lissées, enrichies et/ou normalisées, afin d'améliorer la visualisation ;
- la troisième étape du processus consiste en la définition de la mise en correspondance entre les données et les éléments de la visualisation ;
- l'étape suivante consiste à traduire les données et la mise en correspondance en une représentation permettant l'analyse ;
- la dernière étape est l'analyse de l'utilisateur. Cette étape est cruciale, car c'est à ce moment que l'on permet à l'utilisateur d'ajuster les différents paramètres du processus de visualisation, afin de lui permettre d'explorer les données.

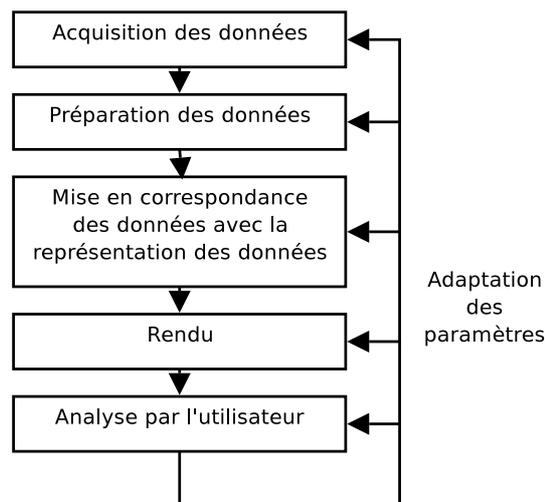


FIG. 3.1 – Processus de visualisation d'information.

3.4 Les difficultés rencontrées dans la visualisation d'information

La conception d'un système de visualisation d'information n'est pas aussi simple qu'elle pourrait le paraître au premier abord. En effet, pour être efficace, le système doit prendre en compte la dimensionalité des données et de la visualisation, ainsi que les limites physiques et psychologiques des sens de l'utilisateur.

3.4.1 Dimensionnalité des données et de la visualisation

Que ce soit pour la représentation de fonctions multidimensionnelles, de nuages de points multidimensionnels ou de dissimilarité, la difficulté de la représentation et la représentativité qui en découle dépendent très fortement de la dimensionnalité intrinsèque des données N (Wijk and van Liere, 1993). En effet, pour $N = 1$, $N = 2$ ou $N = 3$, un simple rendu de graphe ou de volume avec des couleurs en 2D ou en 3D suffit, bien qu'il ne soit pas toujours le mieux adapté. Pour $N = 4$, on peut parfois se ramener aux cas $N = 3$, en considérant qu'une des dimensions est une dimension temporelle, mais, pour $N > 4$, il n'est en général plus possible de représenter l'intégralité des N dimensions sans les déformer. Trois principales techniques sont utilisables, conjointement ou non :

- on fixe la valeur de certaines dimensions, et on représente les autres dimensions (ex : le *scatterplot*, *HyperSlice*, ...),
- on déforme la représentation de manière à tenir compte de l'intégralité des dimensions (ex : *radviz*, les projections, ...),
- on utilise des artifices de représentation pour augmenter le nombre d'attributs visualisables (ex : couleur, forme, direction, ...). L'utilisation de cette dernière technique est réalisée très couramment, mais cela nécessite de prendre des précautions pour qu'elle soit efficace (cf. section 3.4.2).

3.4.2 La perception des sens dans la visualisation d'information

La construction d'une visualisation nécessite de prendre en compte, au maximum, quatre aspects de la perception des sens : la capacité de jugement, la mémorisation, le niveau d'abstraction des représentations et le contexte visuel. La figure 3.2 synthétise ces quatre aspects.

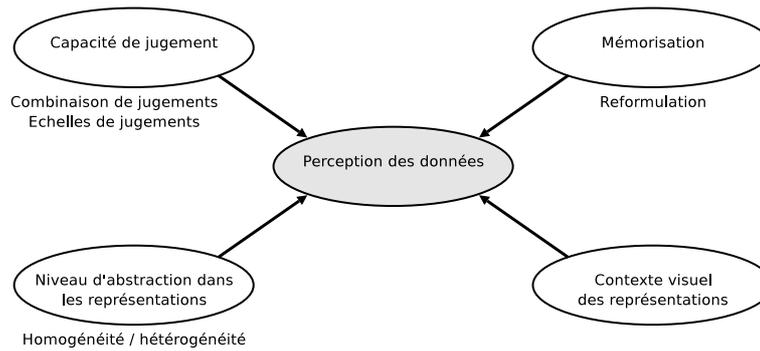


FIG. 3.2 – Les quatres aspects principaux qui influent sur la perception des données.

3.4.2.1 La capacité de jugement

Plusieurs études faites par G. Miller (1956) (Fayyad et al., 2001) ont eu pour objectif l'étude des aptitudes de l'homme à prendre en compte des stimuli. Les études portant sur le jugement absolu de stimuli sonores (pitch, force), gustatifs (salinité) et visuels (positionnement sur une ligne, taille de carrés, longueur, orientation et courbure de lignes) ont permis de montrer que l'homme était capable de distinguer approximativement 10 niveaux différents à chaque fois. G. Miller a également effectué une étude en utilisant plusieurs stimuli en simultané, il a alors constaté une augmentation des capacités de jugement, mais cette capacité n'est pas le produit des capacités des différents stimuli. Ce résultat est confirmé dans (Fayyad et al., 2001) où est évoquée une expérience impliquant la taille, la couleur et la luminosité de carrés de capacités respectives de jugement 5, 8 et 5. La capacité de jugement absolue des stimuli, lorsqu'ils sont combinés, est alors de 17 et non pas $5 \times 8 \times 5 = 200$. (Baker and Wickens, 1995) souligne le fait que les ressources utilisées pour l'analyse auditive et visuelle sont différentes et que les ressources impliquées dans l'analyse des formes, des couleurs et des orientations sont également différentes.

Cleveland (Baker and Wickens, 1995) s'est également intéressé à l'étude de la perception des stimuli. Il est arrivé à la conclusion que la capacité de l'homme à effectuer des jugements relatifs est significativement supérieure à la capacité d'effectuer des jugements absolus de stimuli. De plus, Cleveland indique, par ordre décroissant de capacité, que les tâches consistant à détecter des changements relatifs sont :

- positionnement sur une échelle commune,
- positionnement sur des échelles communes non alignées,
- repérage des longueurs,
- repérage des angles et des inclinaisons (dépend beaucoup de l'orientation et du type de l'objet affiché),
- repérage des aires,
- repérage des volumes,
- repérage des couleurs : teinte, saturation, densité.

D'un autre côté, la loi de Stevens (Fayyad et al., 2001) souligne le fait que l'échelle (visuelle, auditive, ...) est très importante. En effet, pour un stimulus visuel, la perception dans un jugement absolu de la taille pour un attribut graphique linéaire est approximativement la taille réelle élevée à une puissance comprise entre 0.9 et 1.1. Pour une aire, la puissance considérée se situe dans l'intervalle [0.6, 0.9] et celle pour un volume est dans l'intervalle [0.5, 0.8]. Cette loi implique donc que la notion d'échelle absolue est mieux traduite par un attribut graphique linéaire que par une aire ou un volume.

3.4.2.2 La mémorisation et la reformulation

Comme il a été évoqué plus haut, le nombre moyen de stimuli perceptibles par l'homme est approximativement de 10. Pour augmenter le nombre d'informations qu'un individu peut prendre en compte, il est nécessaire de faire intervenir la mémoire immédiate. En moyenne, la mémoire immédiate permet à l'homme de conserver 7 informations (plus ou moins 2). Chaque information peut être de plusieurs types : une valeur simple, un stimulus, une combinaison de stimuli ou quelque chose de plus complexe. Pour permettre à chaque case mémoire de retenir une information complexe, la personne a besoin de recoder les informations qu'elle cherche à mémoriser. Ce recodage peut prendre plusieurs formes (une image, un symbole, une phrase, une association, ...). Il est donc parfaitement possible d'affirmer que dans une visualisation, quelle qu'elle soit, la prise en compte de l'activité de reformulation par l'utilisateur est importante. Cependant, dans (Fayyad et al., 2001), il nous est rappelé que ce recodage des informations est propre à chaque individu et que, par conséquent, des variations tant au niveau de la personne qu'au niveau du contexte dans lequel il est fait sont très probables. Une visualisation doit donc s'efforcer de favoriser cette reformulation, mais elle ne doit pas l'imposer.

3.4.2.3 Le niveau d'abstraction des représentations

D'un autre côté, (Baker and Wickens, 1995) rappellent que la décidabilité sur une représentation visuelle dépend fortement du niveau d'abstraction du symbole utilisé dans la visualisation. La loi de Gestalt établit que les formes fermées, régulières et symétriques impliquent une forte unicité de la perception de la forme (ex : $\langle \rangle$ est perçu comme un unique objet tandis que $\rangle \wedge$ ne l'est pas). Cependant, l'encodage de plusieurs variables dans un unique *glyph*¹ facilite l'intégration des attributs, mais cette intégration rend difficile l'action de focalisation sur une des caractéristiques, tout en ignorant les autres. De plus, la comparaison de deux ensembles de données n'est possible que s'ils sont codés sur un même ensemble de caractéristiques, car l'homogénéité améliore la découverte de lien ou de différence entre les données.

3.4.2.4 Le contexte visuel

L'arrangement spatial dans la visualisation joue également un rôle important (Baker and Wickens, 1995). En effet, si les données ont une organisation géographique intrinsèque, la compréhension est améliorée en la conservant. Cependant ce choix laisse moins d'attributs graphiques utilisables pour les autres attributs. D'autres facteurs permettent d'améliorer la perception d'une visualisation : les ombres fournissent les informations sur la hauteur et la distance au sol, la perspective informe sur la profondeur, les grilles et les marqueurs permettent de clarifier la position des objets de la scène tandis que la stéréoscopie améliore les temps de découverte de connaissances à partir des données. Il est aussi important de ne pas encombrer une visualisation avec trop de détails car l'homme n'est pas capable de tous les appréhender efficacement et cela peut même nuire à la compréhension. Il est donc nécessaire de pouvoir identifier les objets importants par des marqueurs. Il est aussi important de savoir que les textures, les gradients et la vue stéréoscopique ont moins d'importance que la lumière et les ombres dans une représentation.

Finalement, le rôle d'une visualisation d'information consiste à définir une carte mentale des données. Cette carte mentale sert de moyen d'encodage de la position relative des objets les uns par rapport aux autres et de moyen de codage des attributs des objets dans l'environnement.

¹Le terme anglais *glyph* représente une forme en général.

3.5 Les techniques de visualisation

Depuis ses débuts, le domaine de la visualisation d'information n'a cessé d'inspirer de nombreux travaux. Ces travaux ont donné naissance à de nombreuses techniques de représentation de l'information. Certaines d'entre elles ont particulièrement marqué le domaine et ont inspiré de nombreuses variantes plus ou moins adaptées à des domaines particuliers. Dans cette section, nous n'allons pas tenter de dresser un panorama exhaustif des différentes techniques existantes ni des différents logiciels les implémentant. Nous allons nous attacher à la description des principales techniques de visualisation de l'information indépendamment des logiciels ou « pack logiciel » dans lesquelles elles sont implémentées. Dans cette étude, nous avons volontairement ignoré les techniques de visualisation trop spécifiques, telles que la visualisation médicale, la visualisation de volume ou de force, . . . afin de nous concentrer sur la visualisation de graphes, de données multidimensionnelles et de dissimilarité/similarité. Le lecteur intéressé par une bonne introduction aux techniques et problèmes principaux de la visualisation d'information pourra se référer à (Card et al., 1999).

3.5.1 Le graphe de courbes

Historiquement, le graphe de courbes (*line graphe*) est la première visualisation qui a été utilisée. Son principe est simple, il consiste en la représentation habituelle de fonction continue $y = f(x)$. Bien que ne supportant que deux dimensions, cette représentation peut être étendue en superposant plusieurs graphes (Fayyad et al., 2001). Ainsi, en superposant n graphes en utilisant le même paramètre x , on représente un total de $n + 1$ dimensions de l'espace. Ce type de graphe est appelé graphe de courbe multiple (*multiple line graph*) (cf. figure 3.3).

Cependant, comme la dimension x joue un rôle particulier (les valeurs ne se répètent pas et x a le sens de « paramètre » des données), cette particularité peut parfois poser problème lors de l'interprétation. De plus, il peut être nécessaire d'utiliser des échelles différentes en ordonnée et des *offset* de décalages rendant la représentation confuse le plus souvent à partir de trois courbes.

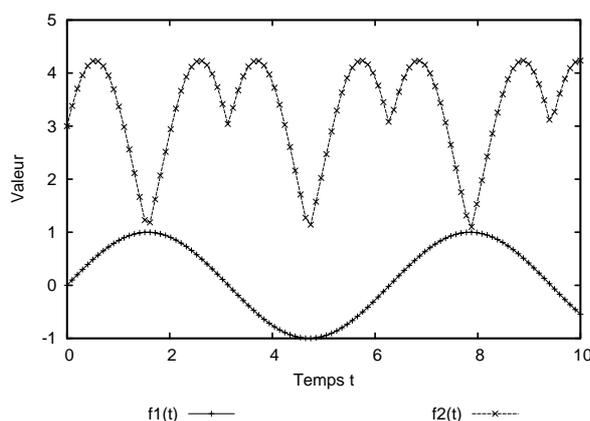
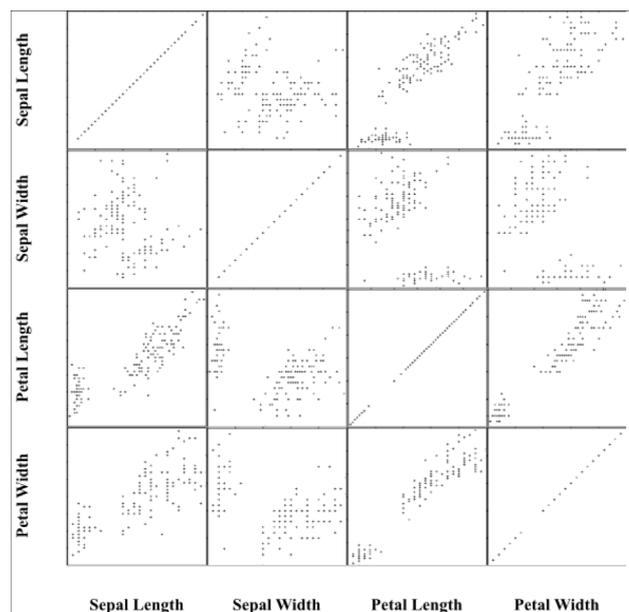


FIG. 3.3 – Un exemple de graphe de courbes.

3.5.2 Le *scatterplot*, la matrice de *scatterplots* et ses dérivées

3.5.2.1 Le *scatterplot*

Le nuage de points ou *scatterplot* est un graphe en deux dimensions affichant un ensemble de couples (x, y) sous la forme d'un nuage de points.

FIG. 3.4 – Exemple de matrice de *scatterplots* sur la base Iris (Fisher, 1936).

3.5.2.2 La matrice de *scatterplots*

La matrice de *scatterplots* (Cleveland, 1993) (Wong and Bergeron, 1997) (cf. figure 3.4) est, comme son nom l'indique, une matrice de plusieurs *scatterplots* dont les axes changent. Si on considère (x_1, x_2, \dots, x_n) les dimensions dans lesquelles sont décrites les données, alors la matrice est obtenue en énumérant les paires de dimensions et en traçant le *scatterplot* associé pour chaque paire (cf. figure 3.4). Comme on peut le remarquer, la matrice est symétrique et la diagonale² haut/gauche vers bas/droite représente l'étalement des valeurs de la coordonnée.

L'un des principaux avantages de la matrice de *scatterplots* est de permettre la représentation de données discrètes sans donner un rôle particulier à une dimension de l'espace (Wijk and van Liere, 1993), contrairement aux graphes de courbes. Les matrices de *scatterplots* sont très populaires pour l'analyse de données, car elles permettent la recherche des corrélations entre deux dimensions, de classe, de tendance et de données incorrectes.

La matrice de *scatterplots* permet également l'utilisation de brosses (*brushing*³), la coloration des classes, le zoom, le *jittering*⁴. Elle a aussi été utilisée avec des *glyphs*, des icônes et des couleurs (Fayyad et al., 2001).

Bien que le nombre de dimensions utilisables avec une matrice de *scatterplots* soit limité par la taille de l'écran, il est possible d'en augmenter la capacité de représentation en utilisant une représentation 3D (Scullin et al., 1995), des animations, des couleurs, des formes et l'interaction, afin de montrer pour chaque *scatterplot* jusqu'à 3, 4, 5 ou plus de dimensions (Fayyad et al., 2001).

²Dans certaines implémentations, les *scatterplots* ne sont pas disposés de la même façon. La diagonale est alors du coin haut/droite vers le coin bas/gauche. Par exemple, dans la technique de la matrice de *scatterplots* pseudo-euclidienne que nous proposons au chapitre 6, nous adoptons cette disposition alternative.

³L'utilisation de brosses ou *brushing* est le terme consacré aux techniques permettant la sélection de certains points et le filtrage interactif de ces derniers. (Martin and Matthew O. W., 1995) présente plusieurs modes de *brushing*, implémentés dans le logiciel XmdvTool, pour la matrice de *scatterplots*, les *parallel coordinates* et le *dimensional stacking*.

⁴Le *jittering* est la technique consistant à ajouter un bruit aléatoire aux projections de manière à rendre visible des points ayant les mêmes coordonnées de projection.

3.5.2.3 La technique HyperSlice

La représentation en HyperSlice (Wijk and van Liere, 1993) est une méthode de visualisation dérivée des matrices de *scatterplots* afin de permettre la visualisation de fonction scalaire de plusieurs variables. Par fonction scalaire, on entend les fonctions du type $f(x) = y$ avec x une donnée multidimensionnelle et y un réel. Pour adapter la représentation en *scatterplot*, les auteurs définissent un point courant c de l'espace et un ensemble d'intervalles sur chacune des dimensions autour de c . Chaque *scatterplot* de la matrice est obtenu en sélectionnant deux dimensions x_i et x_j et en dessinant en niveaux de gris la valeur de la fonction en faisant varier x_i et x_j dans les intervalles spécifiés et en laissant les autres dimensions aux valeurs des dimensions du point c .

3.5.3 Les techniques de projection

3.5.3.1 La technique Radviz

La méthode de visualisation *Radviz* (pour visualisation radiale) (Hoffman et al., 1997) (Brunsdon et al., 1998) est une technique de projection utilisant la notion d'ancre de dimensions, comme expliqué dans (Hoffman et al., 1999).

La technique *RadViz* consiste à définir sur un cercle unitaire m points d'ancrage correspondant chacun à une des m dimensions des données, de manière à ce qu'ils soient régulièrement espacés. La projection d'une donnée sur le plan correspond au point d'équilibre annulant la somme des forces exercées sur ce point vers les points d'ancrage. La force s'exerçant entre le point d'équilibre et un point d'ancrage a pour direction le vecteur ayant pour origine le point d'équilibre et pour extrémité le point d'ancrage (cf. figure 3.5 (a)). La force s'exerçant est proportionnelle à la valeur prise par la donnée sur cette dimension.

Le point d'équilibre u_i relatif à la donnée $x_i = (x_{i,1}, \dots, x_{i,m})$ pour les points d'ancrage $S_{j/j=1..m}$ est la solution de l'équation

$$\sum_{j=1..m} (S_j - u_i)x_{i,j} = 0$$

c'est-à-dire

$$u_i = \sum_{j=1..m} \frac{x_{i,j}}{\sum_{k=1..m} x_{i,k}} S_j$$

Ce point d'équilibre est donc simplement le barycentre des points d'ancrage avec pour masse

$$\frac{x_{i,j}}{\sum_{k=1..m} x_{i,k}}$$

De plus, cette représentation assure que, si $x_{i,j} \geq 0$, alors les projections des points sont contenues dans le polygone formé par les points d'ancrage (Brunsdon et al., 1998). Cette condition est facilement réalisable par une étape préliminaire de normalisation des données. La figure 3.5 (b) montre le résultat de la projection d'un jeu de données.

Cette méthode possède pour principal inconvénient sa dépendance à l'ordre des points d'ancrage sur le cercle unitaire. En effet, il est tout à fait possible qu'un ordre particulier fasse que des forces s'annulent mutuellement si leurs points d'ancrage sont diamétralement opposés sur le cercle. En éliminant les configurations d'ancres similaires, à une rotation et une symétrie près, il existe exactement $\frac{(m-1)!}{2}$ projections différentes envisageables (Brunsdon et al., 1998).

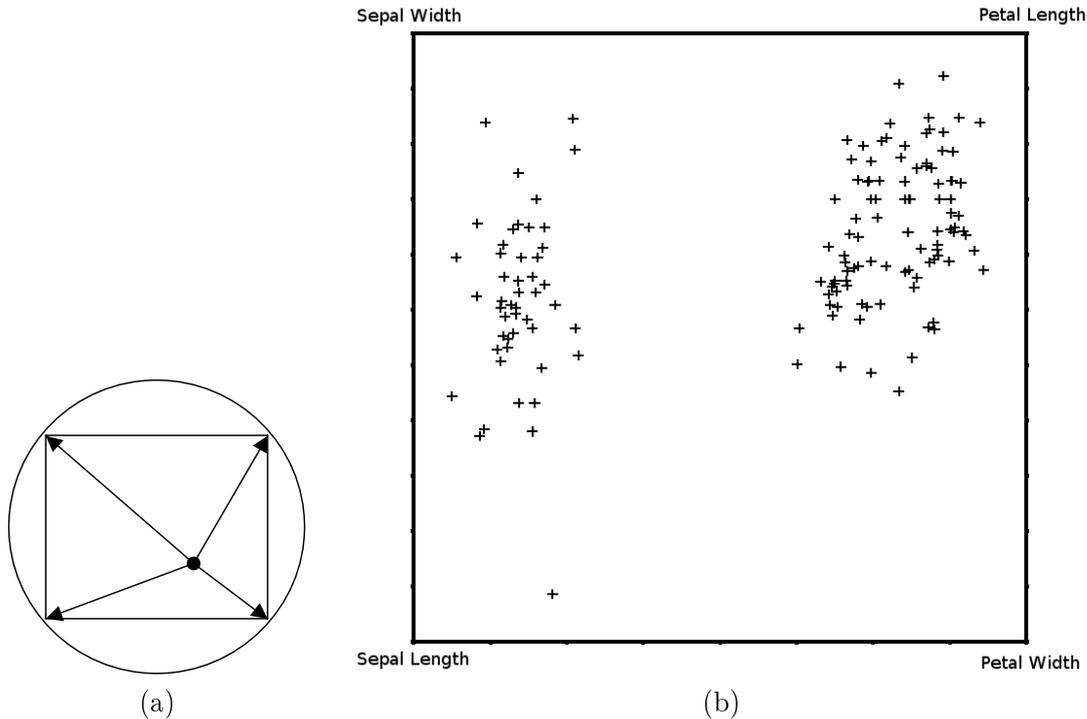


FIG. 3.5 – Visualisation par la technique *RadViz* : (a) les forces s'exerçant sur un point d'équilibre avec 4 ancres, (b) Exemple de projection de la base Iris (Fisher, 1936).

3.5.3.2 Les cartes auto-organisatrices de Kohonen

Les *Self Organizing Maps* (SOM) (Vesanto, 1997) (Vesanto, 1999), autrement nommées cartes auto-organisatrices de Kohonen, du nom du professeur qui les a développées au début des années 80, sont fondées sur un apprentissage neuronal non supervisé et concurrentiel des données. De par leurs organisations spatiales, les SOM sont des outils classiques de projection et de quantification⁵ simultanées.

Les SOM. sont formées, pour des données multidimensionnelles en n dimensions, de neurones uniformément répartis, principalement, sur une grille 1D ou 2D. Chacun des neurones est muni d'un vecteur de n poids. Les neurones sont reliés aux neurones constituant leurs voisinages (plusieurs types de voisinages peuvent être considérés, comme par exemple des voisinages rectangulaires, hexagonaux, ...). Le nombre de neurones et la taille du voisinage déterminent la finesse de l'analyse. Après une initialisation des vecteurs poids aléatoirement, par échantillonnage des données, par des vecteurs propres ou par toutes autres méthodes, les données sont apprises par le réseau. A chaque itération de l'apprentissage, on choisit aléatoirement un vecteur x parmi les données et on calcule sa similarité avec les vecteurs poids des neurones. On détermine alors le neurone ayant les poids les plus similaires à la donnée (*Best Matching Unit* ou BMU). On met alors les vecteurs poids de la BMU et du voisinage à jour en suivant l'équation suivante (Vesanto, 1999) :

$$m(t+1) = m(t) + h(t)(x - m(t))$$

en désignant par $m(t)$ le vecteur poids à mettre à jour, x la donnée, $h(t)$ la fonction noyau du voisinage pour le vecteur. La fonction $h(t)$ est décroissante en fonction du temps t et de la distance du neurone à la BMU. Cet apprentissage a pour effet d'attirer les neurones de la BMU et du voisinage vers la donnée. La figure 3.6 en schématise le principe.

⁵Définition des données en fonction d'un *codebook* discret.

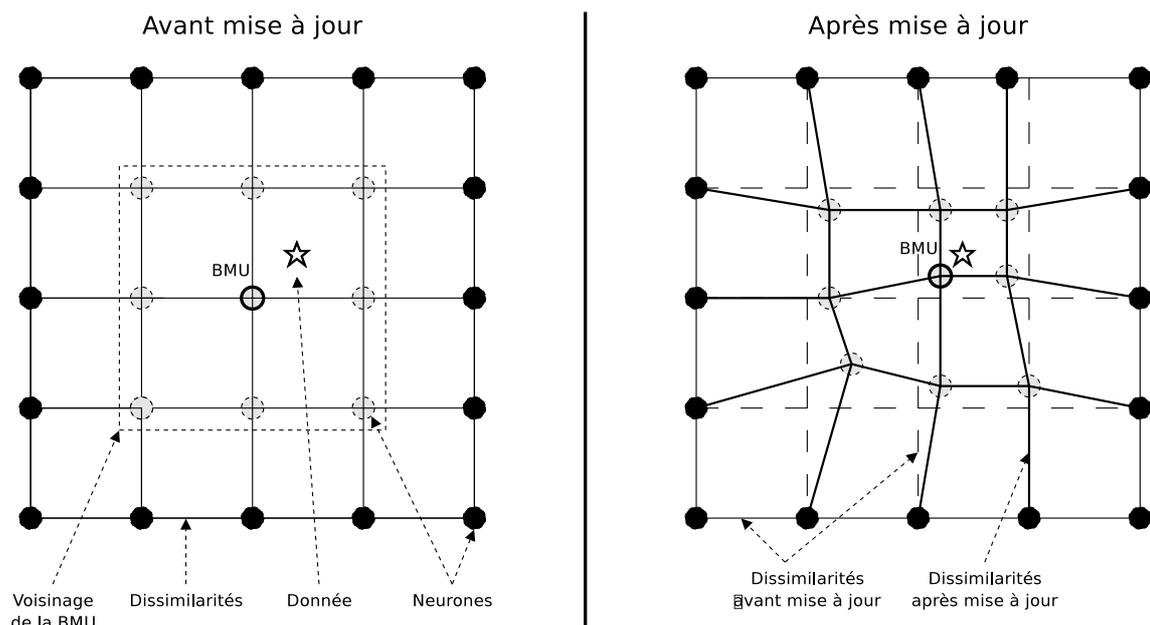


FIG. 3.6 – Schématisation de l’attraction des neurones vers une donnée lors d’une itération de l’algorithme des cartes auto-organisatrices (figure adaptée à partir de (Vesanto, 1997)).

Comme nous l’avons déjà dit plus haut, les SOM réalisent une quantification et une projection des données, c’est-à-dire qu’elles cherchent de bons vecteurs pour servir de représentants des données et en plus elles ordonnent ces vecteurs sur la grille. Les SOM ont également la propriété d’éliminer le bruit dans les données (Vesanto, 1999). Cependant, contrairement aux méthodes de projection standard, les SOM ne conservent pas les distances entre les données directement, elles se contentent de conserver la structure locale des données et, par conséquent, l’interprétation des SOM ne doit se faire que localement.

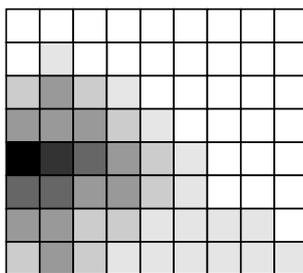


FIG. 3.7 – Exemple de visualisation d’une composante des vecteurs associés aux neurones d’une carte auto-organisatrice en niveaux de gris.

La visualisation des SOM revêt plusieurs aspects :

- la visualisation des composantes des vecteurs de poids des neurones. On visualise en niveaux de gris la valeur de chaque composante sur une carte comme dans la figure 3.7 (Vesanto et al., 1998). Il est de plus possible de rechercher des corrélations entre les diverses composantes des vecteurs de poids en adoptant une visualisation simultanée de plusieurs composantes.
- la visualisation de l’organisation des groupes (*clusters*) dans les SOM. On visualise une matrice de similarité des vecteurs poids avec leur voisinage en niveau de gris.
- la visualisation de la projection des vecteurs de poids. On effectue une projection des

vecteurs poids des neurones par une projection en 2D ou 3D préservant les distances (cf. section 3.5.3.3 sur le *Multidimensional Scaling*), comme la projection de Sammon, et on relie les neurones voisins. Ce type de visualisation permet d’appréhender la structure de la SOM.

- une autre visualisation consiste à comptabiliser le nombre de fois qu’un neurone est choisi comme BMU par les données sous forme de rectangle de taille correspondante ou d’histogramme.
- lorsque les données ont un ordre séquentiel naturel, on peut visualiser la séquence des BMU en les reliant.

Le lecteur désirant voir d’avantage de représentations graphiques des SOM pourra se référer à (Vesanto, 1997) et (Vesanto, 1999).

3.5.3.3 *Multidimensional scaling*

Le multidimensional scaling (MDS) (Borg and Groenen, 1997) (Cox and Cox, 2001), rarement nommé « échelonnement multidimensionnel » en langue française, est un ensemble de techniques qui consistent à projeter un nuage de points connus par leurs proximités respectives dans un espace donné. Le MDS peut à la fois être considéré comme un pré-traitement et comme une méthode de visualisation. On parlera de pré-traitement lorsqu’il s’agit de projeter le nuage dans un espace de dimension quelconque, et de visualisation si la projection s’effectue dans un espace de faible dimension (ex : 1D, 2D ou 3D) avec l’objectif de voir le nuage. Pour effectuer cette projection, le MDS cherche à minimiser une fonction de *stress* dépendant des proximités *a priori* entre les données ($p_{i,j}$) et des proximités effectives de leurs représentants dans l’espace projeté ($d_{i,j}$). De nombreuses fonctions de *stress* ont été étudiées au fil des années (Borg and Groenen, 1997) (Cox and Cox, 2001). Les techniques fondamentales sont décrites dans le tableau 3.1.

Nom	Objectif
classical MDS	$p_{i,j} = d_{i,j}$
interval MDS	$a + bp_{i,j} = d_{i,j}$
ordinal MDS	$p_{i,j} < p_{k,l} \implies d_{i,j} \leq d_{k,l}$
MDS généralisé	$f(p_{i,j}) = d_{i,j}$ en désignant par f une fonction réelle monotone

$d_{i,j}$ peut être définie de plusieurs façons. De manière classique, $d_{i,j}$ est une distance de Minkowski ($d(x, y) = (\sum_t |x_t - y_t|^K)^{1/K}$) mais elle peut également être une distance sur un cercle ou toute autre mesure.

TAB. 3.1 – Techniques fondamentales de *multidimensional scaling*.

3.5.4 La visualisation de graphes

La visualisation de graphes représente tout un pan de la visualisation d’information. En effet, de nombreuses techniques y sont dédiées. Les différentes visualisations se distinguent les unes des autres suivant quatre critères : le type de graphe représenté (arbre, graphe complet, orienté, non orienté, . . .), la taille du graphe, le placement des noeuds du graphe et la représentation des noeuds et des arcs entre les noeuds.

La technique *Nicheworks* (Wills, 1999) en est un bon exemple. Cette technique est principalement destinée à la représentation d’arbres. Pour cela, elle adopte une stratégie de division⁶ récursive de l’espace en fonction de la taille des sous-arbres (cf. figure 3.8).

⁶Il s’agit d’une division d’angles.

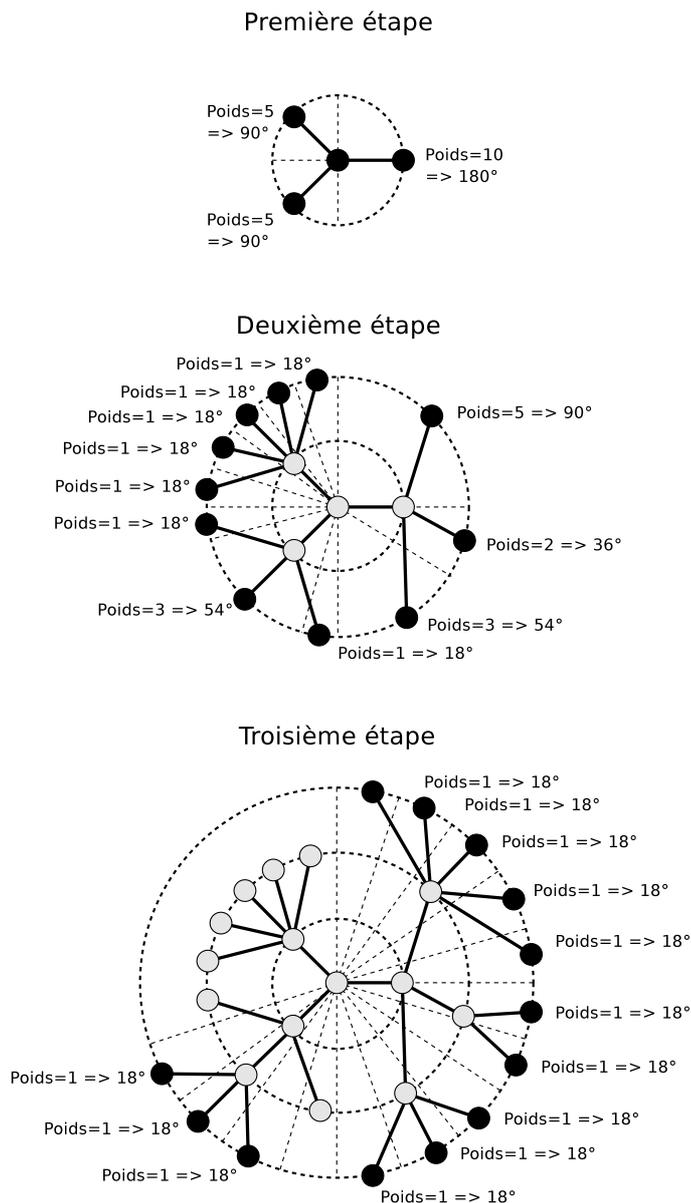


FIG. 3.8 – Principe de positionnement/partitionnement de la technique *Nicheworks* (figure adaptée de (Wills, 1999)).

Étant donnée la taille de ce sous-domaine de la visualisation d'information, nous n'allons pas effectuer un inventaire des principales techniques de la visualisation de graphes. Cependant, nous encourageons le lecteur intéressé à se référer aux articles (Di Battista and Eades, 1994) et (Herman et al., 2000) établissant une bonne revue des techniques existantes dans le domaine. D'autres techniques connues, telles que les *Tree-maps* (Johnson and Shneiderman, 1991) pour des hiérarchies ou la technique du *Cone-tree* (Robertson et al., 1991), y sont référencées.

3.5.5 Les matrices de similarité en niveaux de gris

Les matrices de similarité en niveaux de gris ou *shaded similarity matrix*, également connues sous les noms de *shaded proximity matrix* et *trellis diagram*, sont un moyen de visualiser des similarités entre des éléments. Les similarités sont matérialisées par un niveau de

couleur variant du plus clair au plus foncé, respectivement pour les faibles similarités et les fortes similarités (Wang et al., 2002b) (Wang et al., 2002a). La figure 3.9 donne un exemple d'une telle matrice.

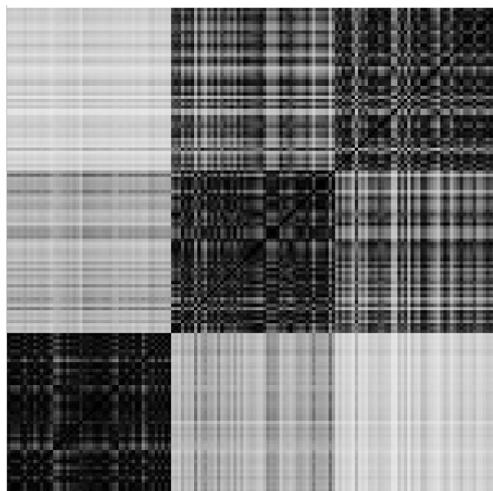


FIG. 3.9 – Exemple de matrice de similarité en niveaux de gris de la base Iris (Fisher, 1936). Les niveaux de gris sombres marquent une forte similarité entre les données.

Cette représentation des données est très efficace pour détecter des groupes de données, ces derniers apparaissant alors sous forme de carrés foncés et symétriques sur la diagonale. Comme il existe de nombreuses façons d'arranger les éléments pour la représentation ($n!$ pour n objets), on utilise généralement une méthode d'organisation des objets dans le but de faire ressortir les caractéristiques des données. Plusieurs types de méthodes existent. Beaucoup cherchent à trouver une organisation des données minimisant un critère, comme la variation locale des niveaux de similarité. D'autres appliquent une organisation du type *concept tree based ordering* consistant à construire un arbre de décision, et à organiser les données afin que la représentation respecte l'arbre de décision (Wang et al., 2002b). Ce type d'organisation peut être préférable, car la structure sous-jacente de l'arbre de décision rend l'interprétation de la matrice plus facile.

3.5.6 *Parallel Coordinates* et ses variantes

3.5.6.1 *Parallel Coordinates*

La technique des *parallel coordinates* (fig. 3.10), initialement introduite par A. Inselberg au début des années 1980, consiste à représenter chaque point du jeu de données par une polyligne⁷ (Inselberg and Dimsdale, 1990) (Inselberg, 2002). Pour cela, on considère n axes parallèles (n est la dimension des données à représenter) régulièrement espacés. Chaque axe correspond à une dimension des données. Un point des données est alors représenté par la polyligne reliant les différents axes. L'intersection de la polyligne avec un axe correspond à la valeur de la dimension associée à l'axe pour cette donnée. Si les axes parallèles sont aux coordonnées d'abscisse $1..n$, alors le point $x = (x_1, x_2, \dots, x_n)$ est représenté par la polyligne passant par les points de coordonnées $(1, x_1), (2, x_2), \dots, (n, x_n)$. Cette méthode transforme donc chaque point de \mathbb{R}^n en $(n - 1)$ segments joignant n lignes parallèles.

⁷Une polyligne est le résultat de la mise bout à bout de plusieurs segments.

Cette transformation des points possède des propriétés remarquables très intéressantes (Inselberg and Dimsdale, 1990) (Chou et al., 1999) (Inselberg, 2002) :

- une droite en dimension n peut s'écrire sous la forme de $n - 1$ équations $x_{i+1} = m_i x_i + b_i \forall i = 1..n - 1$;
- les points situés sur une droite de \mathbb{R}^n sont transformés en un ensemble de polygones dont chaque groupe de segments s'intersecte au point de coordonnées $\left(\frac{i}{1-m_i}, \frac{b_i}{1-m_i}\right) \forall i = 1..n - 1$ si $m_i \neq 1$;
- quand $m_i = 1$, la droite d'angle directeur $\frac{b_i}{i}$ joignant un point situé à l'infini caractérise la relation linéaire (ce n'est plus un point d'intersection mais une direction) ;
- la rotation ou la translation d'une ligne en dimension n par rapport à un point se traduit respectivement par une translation et une rotation des points d'intersections des groupes de segments ;
- des lignes parallèles en dimension n sont alors transformées en polygones dont les points d'intersections sont alignés verticalement ;
- par conséquent, un hyperplan en dimension n est représenté par plusieurs alignements verticaux de points d'intersection.

Cette méthode supporte classiquement un certain nombre d'interactions et d'outils (Hauser et al., 2002) :

- le réordonnancement des axes,
- le renversement des axes *i.e.* le retournement vertical de l'axe,
- l'utilisation d'une normalisation locale ou globale pour chacun des axes,
- le changement de la longueur des axes,
- l'ajout et la suppression d'axes,
- les détails sur une donnée à la demande,
- les brosses de sélection (*brushing*) :
 - les brosses angulaires permettant la sélection et le filtrage de points grâce à un critère de relation d'ordre sur deux de ses coordonnées.
 - les brosses composites sont obtenues par combinaison via des opérateurs logiques d'autres brosses. Elles peuvent être à valeur booléenne ou continue, grâce à l'utilisation d'opérateurs de logique floue.
- les histogrammes des valeurs prises sur les axes permettent de mieux appréhender la distribution des données par rapport à chaque axe. Ces histogrammes sont affichés en semi-transparence, afin de ne pas perturber trop la relation visuelle décrite par les polygones.

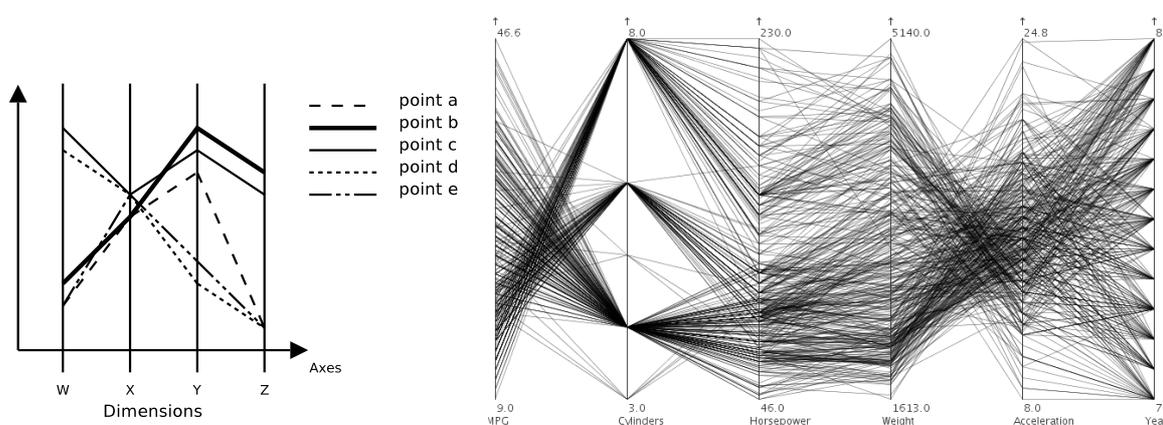


FIG. 3.10 – Deux exemples d'applications de la technique de *Parallel Coordinates*. Le deuxième exemple a été obtenu à l'aide du logiciel Parvis (Ledermann, 2002) sur la base de données *Car* (Newman et al., 1998).

Un certain nombre d'extensions de la technique des *Parallel Coordinates* ont été développées.

3.5.6.2 3D Parallel Coordinates

3D Parallel Coordinates (Mroz et al., 1998) : une généralisation des *Parallel Coordinates* dans laquelle les axes parallèles sont remplacés par des plans parallèles. Une donnée est représentée par une polyligne joignant des plans parallèles. Chacun des plans correspond à deux coordonnées. Par exemple, si deux plans parallèles représentent les dimensions (x_1, x_2) et (x_3, x_4) , le point $(0, 0, 1, 1)$ sera représenté par le segment joignant le point de coordonnées $(0, 0)$ du premier plan et le point de coordonnées $(1, 1)$ du deuxième plan.

3.5.6.3 Hierarchical Parallel Coordinates

Hierarchical Parallel Coordinates (Fua et al., 1999) : dans cette extension, les données sont d'abord classées au sein d'une hiérarchie. Les représentants des classes à un certain niveau de la hiérarchie sont visualisés grâce à la technique des *Parallel Coordinates*. Afin de représenter la taille et la variance des classes, les polygones sont élargies à une largeur dépendante de la taille de la classe. Ces polygones deviennent transparentes au fur et à mesure que l'on s'éloigne du représentant de la classe.

3.5.6.4 Circular Parallel Coordinates

Circular Parallel Coordinates (Hoffmann and Grinstein, 2005) : cette extension correspond à un positionnement en étoile des axes parallèles de la technique de *Parallel Coordinates* (cf. figure 3.11). L'intérêt principal de cette technique est que chaque axe joue le même rôle.

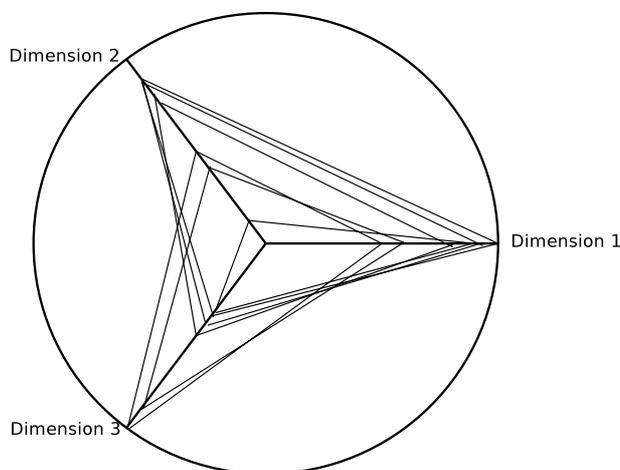


FIG. 3.11 – Exemple d'utilisation de la technique de *Circular Parallel Coordinates* sur un jeu de données fictifs.

3.5.7 L'empilement de dimensions

L'empilement de dimensions ou *dimensional stacking* est une technique générale permettant de représenter sans perte d'information des données multidimensionnelles tout en minimisant les occlusions. Pour cela, une stratégie récursive est adoptée dans la construction de la représentation. Plusieurs variations sur le principe du *dimensional stacking* ont été réalisées dans la littérature. Nous allons décrire rapidement les trois plus connues : *N-Land*, *co-plot* et *Worlds within Worlds*.

3.5.7.1 *N-land*

La technique *N-land* (Ward et al., 1994) considère des données à valeurs discrètes codées sur un nombre impair de dimensions⁸. On applique récursivement la méthode suivante : on choisit chacune des dimensions l'une après l'autre, en considérant que la dimension choisie dépend uniquement des dimensions précédentes. En prenant les dimensions deux par deux, on construit alors un graphe contenant pour chaque coupe de coordonnées, un autre graphe. Ce graphe correspond aux deux dimensions suivantes et ce processus est itéré jusqu'à épuisement de toutes les dimensions. La dernière dimension est représentée par un niveau de gris de manière à ce que chaque point corresponde aux coordonnées imbriquées de tous les graphes. La figure 3.12 schématise ce processus. Cette technique permet de détecter et d'analyser des relations linéaires impliquant jusqu'à 4 dimensions.

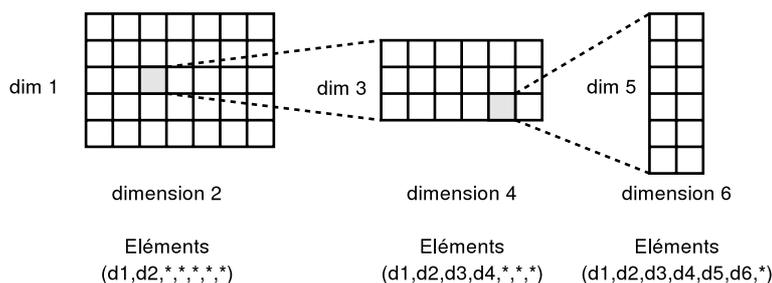


FIG. 3.12 – Structure d'imbrication pour la construction d'une représentation *N-land* pour 7 dimensions.

3.5.7.2 *Co-plot*

La technique de *co-plot* est similaire au *dimensional stacking*. Dans (Carr, 1995), l'auteur utilise un *co-plot* pour rechercher un minimum local d'une fonction de 4 variables. La technique de *co-plot* consiste à définir une grille sur 2 variables, et dans chacune des zones de ces grilles à dessiner le résultat de la fonction en faisant varier les 2 autres variables. Dans l'article, les valeurs de la fonction sont représentées par des angles formés par un petit segment par rapport au point d'ancrage. C'est dans cette variation de la présentation du résultat que se trouve la différence avec *N-Land*.

3.5.7.3 *Worlds within Worlds*

Le principe de la technique *Worlds within Worlds* (Feiner and Beshers, 1990) (Wong and Bergeron, 1997) est simple. Pour l'expliquer, prenons un exemple. On considère une fonction f de six variables $f(x_1, x_2, x_3, x_4, x_5, x_6)$. Représenter les valeurs prises par une telle fonction est difficile. Un moyen simple consiste à fixer trois variables parmi les six et à représenter le nuage de points correspondants. C'est ce que fait la technique *Worlds within Worlds*. Elle considère un graphe principal en trois dimensions. A l'aide d'un moyen de contrôle quelconque, l'utilisateur choisit un point dans ce sous-espace. Une fois ce point choisi (et donc après avoir fixé les paramètres correspondants de la fonction), un nouveau graphe en trois dimensions est dessiné, permettant le choix de la valeur de trois nouvelles variables. Le processus se renouvelle jusqu'à épuisement des variables de la fonction. Le dernier graphe peut permettre de représenter des courbes, des surfaces ou un nuage de points en fonction des propriétés de f . A tout moment, l'utilisateur peut choisir de déplacer le point de référence dans chacun

⁸Si la dimension des données est paire, il suffit alors d'ajouter une dimension fictive aux données et de fixer la même valeur pour toutes les données.

des graphes 3D intermédiaires. L'avantage principal de cette représentation est de permettre une représentation exacte des données, cependant elle reste difficile à utiliser en raison de la complexité intrinsèque des fonctions f . Dans N-vision (Beshers and Feiner, 1993), les auteurs ont ajouté un système de règles (AutoVisual) pour faciliter l'utilisation de cette représentation, en assistant l'utilisateur quant au choix des variables. La figure 3.13 montre N-vision en action.

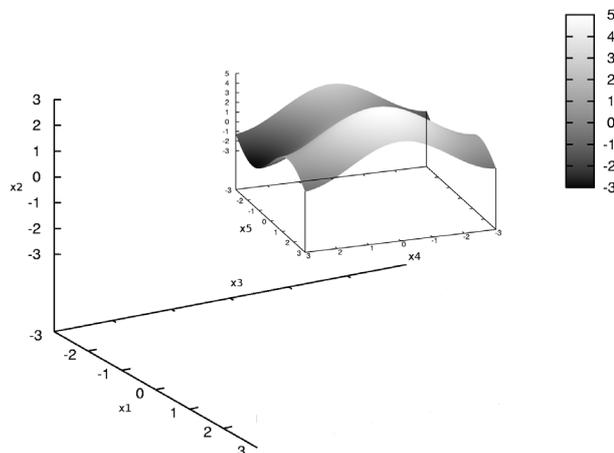


FIG. 3.13 – Exemple d'utilisation de la technique *Worlds within Worlds* pour la fonction $f(x_1, x_2, x_3, x_4, x_5) = x_1 * x_2 + x_3 * (\cos(x_4) + \sin(x_5))$.

3.5.8 Visualisations iconiques

Une visualisation iconique est une visualisation dans laquelle chaque donnée correspond à un élément visuel bien identifié et dans laquelle chaque élément visuel est obtenu à partir de primitives ou de couleurs paramétrées par les coordonnées du point. Ces éléments visuels peuvent être de différents types.

3.5.8.1 Les visages de Chernoff

Dans la visualisation à l'aide des visages de Chernoff (Chernoff, 1973), les éléments visuels sont des visages. Les primitives correspondent, par exemple, à l'ovale de la tête, la taille des yeux ou la longueur du nez, . . . La figure 3.14 montre le résultat de cette technique.

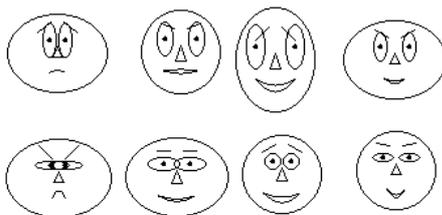


FIG. 3.14 – Visualisation d'un jeu de données fictif avec les visages de Chernoff. Cette figure a été construite grâce au logiciel *Chernoff faces in Java* (Wiseman, 1998).

3.5.8.2 *Star glyphs*

Dans la technique des *stars glyphs* (Siegel et al., 1972) (Chambers et al., 1983), les primitives correspondent à des segments d'origine le centre du star glyph. L'angle formé par deux segments adjacents est constant, tandis que la longueur de chaque segment dépend de la valeur prise par la coordonnée de chacune des données. Les extrémités des segments sont, en général, jointes de manière à fermer les figures. La figure 3.15 montre le résultat de cette technique.

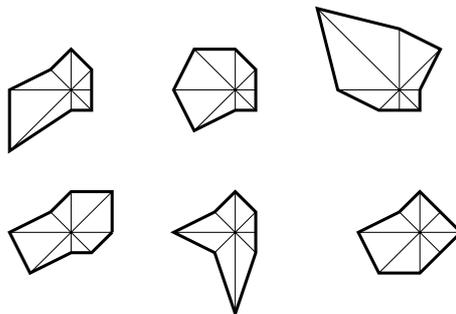


FIG. 3.15 – Visualisation d'un jeu de données fictif avec la technique des *stars glyphs*.

3.5.8.3 *Stick figures*

La technique des stick figures (Pickett and Grinstein, 1988) (Grinstein et al., 1989) est une approche similaire aux visages de Chernoff. Au lieu de paramétrer les différentes composantes d'un visage, la technique paramètre un enchaînement de segments (le nombre de segments et la structure pouvant varier selon les implémentations) : la longueur de chacun des segments et les angles formés sont des primitives graphiques. A la différence des visages de Chernoff, les sticks sont destinés à un affichage dense de points. Cet affichage compose alors une texture servant d'indicateur visuel sur les données. (Keim, 1997) présente un exemple d'application de cette technique.

3.5.8.4 Les courbes d'Andrews

Les courbes d'Andrews (Andrews, 1972) peuvent être considérées comme des visualisations iconiques. En effet, dans le premier cas, au lieu d'être un opérateur de positionnement spatial, l'opérateur de combinaison des primitives graphiques est une somme. Au lieu d'être des éléments graphiques, les primitives sont des fonctions. Pour cela, l'élément visuel associé à une donnée x de coordonnées (x_1, x_2, \dots, x_N) est la fonction définie par

$$f_x(t) = x_1 \frac{1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + x_4 \sin(2t) + x_5 \cos(2t) + \dots$$

La représentation consiste donc à dessiner les différentes fonctions f_x en faisant varier t dans l'intervalle $]-\pi, \pi[$. De par ce mode de représentation, la visualisation peut être considérée comme une visualisation de la transformée de Fourier des données (Hoffmann and Grinstein, 2005).

3.5.9 Techniques à base de pixels

Les techniques à base de pixels sont des visualisations cousines des visualisations iconiques. En effet, dans ces visualisations, chaque composante de chaque donnée est représentée par un pixel de couleur. Mais, contrairement aux visualisations iconiques, les différents pixels

associés à une donnée ne sont pas regroupés dans l'espace. En lieu et place, les pixels correspondant à une même composante sont agencés ensemble dans la représentation. On trouve principalement deux techniques réalisant cette métaphore : les *circle segments* et VisDB.

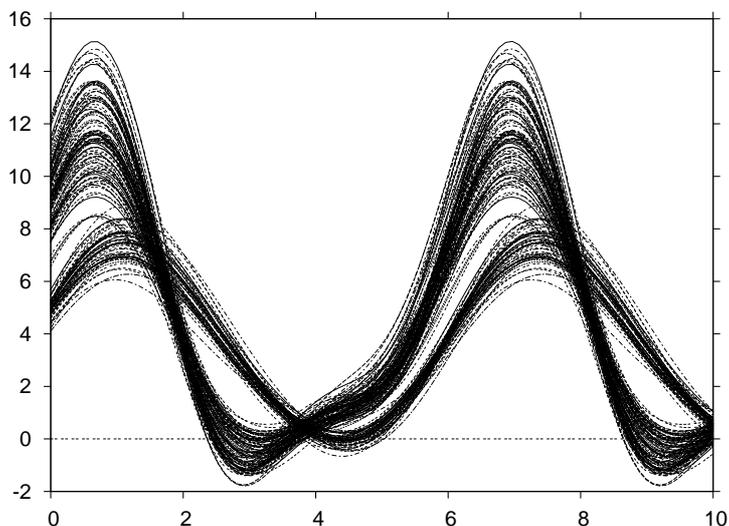


FIG. 3.16 – Visualisation de la base de données Iris (Fisher, 1936) selon la technique des courbes d'Andrews.

3.5.9.1 *Circle segments*

La technique de segment de cercle est une technique orientée pixel consistant à définir, pour chaque dimension des données, une portion angulaire d'un disque. Chacune des portions est de la même taille que les autres. On considère alors les données après les avoir triées dans un ordre donné (habituellement la valeur d'une dimension). Les valeurs des dimensions sont dessinées séparément suivant cet ordre dans chacune des portions. Pour une dimension donnée et par conséquent pour sa portion de disque, les valeurs prises sont représentées du centre du disque vers l'extérieur par remplissage successif d'un côté à l'autre et inversement (Ankerst et al., 1996) (Ankerst, 2001) (Ankerst, 2000). La couleur de chaque point indique la valeur prise par l'attribut. Le principe de remplissage est schématisé par la figure 3.17 (a). La figure 3.17 (b) montre le résultat de la technique sur un jeu de données importants.

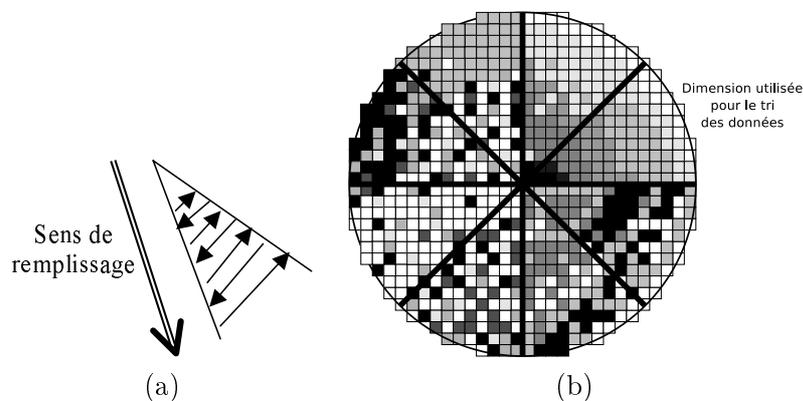


FIG. 3.17 – Visualisation avec les *circle segments* : (a) principe de remplissage d'une portion pour les *circle segments*, (b) exemple d'application de la technique de *circle segments* pour des données fictives en 8 dimensions.

3.5.9.2 VisDB

L'approche suivie par VisDB (Keim and Kriegel, 1994) (Ward and Keim, 1997) est similaire à celle suivie par les *circle segments*. Cependant, les pixels sont arrangés dans des rectangles juxtaposés les uns aux autres (cf. figure 3.19) au lieu d'être arrangés dans des portions de disques. La technique VisDB a pour objectif la visualisation du résultat d'une requête dans une base de données. Par conséquent, un attribut supplémentaire est ajouté aux données. La valeur de cet attribut mesure l'adéquation de la donnée à la requête.

L'ordre des pixels dans les rectangles est défini par la mesure d'adéquation de la donnée à la requête. Le positionnement est effectué en suivant une spirale à partir du centre du rectangle (cf. figure 3.18). Les colorations des pixels, aussi bien pour la mesure d'adéquation de la donnée à la requête que pour les dimensions des données, sont définies et ajustées par l'utilisateur. La visualisation simultanée de toutes les dimensions (cf. figure 3.19) permet de détecter les liaisons dans les données.

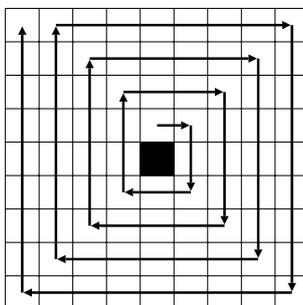


FIG. 3.18 – Visualisation du mode d'arrangement des pixels par la technique VisDB (Schéma adapté de (Keim and Kriegel, 1994)).

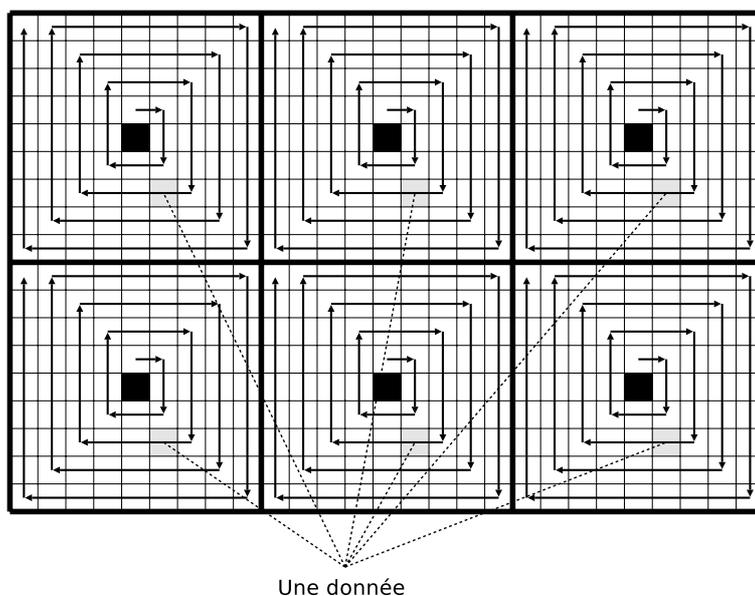


FIG. 3.19 – Visualisation simultanée de plusieurs dimensions par la technique VisDB (Schéma adapté de (Keim and Kriegel, 1994)).

3.6 Conclusion

Les techniques que nous venons d'évoquer ne représentent qu'une petite partie des techniques de visualisation d'information. Cependant elles regroupent, d'une façon ou d'une autre, l'ensemble des briques de base actuellement utilisées pour la construction de nouvelles représentations.

Lorsque l'on consulte la littérature du domaine, on se rend rapidement compte que, malgré le nombre assez impressionnant de visualisations disponibles dans la littérature, aucune brique de base véritablement originale n'est apparue depuis plusieurs années. En effet, les recherches actuelles ne font qu'essayer d'améliorer et d'agencer les briques existantes, en les appliquant à des problèmes spécifiques. De plus, les « nouvelles » visualisations ne sont réellement exploitées que par leurs créateurs. Une partie des recherches actuelles a donc plutôt tendance à s'orienter vers la conception d'un système logiciel complexe, intégrant de multiples visualisations, capables d'interagir les unes avec les autres, afin de les rendre plus accessibles aux utilisateurs.

Dans le cadre de cette thèse et notamment du chapitre 6, nous ne nous occuperons que de la définition d'une nouvelle technique de visualisation de dissimilarité, sans nous préoccuper de son intégration avec d'autres visualisations, que nous réservons pour des travaux futurs.