

# Visualisation interactive d'information

Mountaz Hascoët et Michel Beaudouin-Lafon

Laboratoire de Recherche en Informatique, UMR8623 du CNRS  
LRI, bâtiment 490, Université Paris-Sud, 91405 Orsay Cedex  
mountaz@lri.fr, mbl@lri.fr

## Résumé

Longtemps réservée aux bibliothécaires ou autres professionnels de la documentation, la recherche d'information est aujourd'hui devenue une activité courante pour un grand nombre d'utilisateurs. Les interfaces graphiques de recherche relèvent presque toutes d'un modèle unique et primaire : un champ de saisie uniligne et un bouton permettant de valider la requête saisie dans le champ. Face à la diversité des individus et de leurs besoins en informations, de nombreuses techniques de visualisation et d'interaction innovantes ont émergés ces dix dernières années. Pourtant ces innovations sont rarement utilisées et le but de cet article est de faire le point sur l'état de l'art des techniques de visualisation et d'interaction qui peuvent être pertinentes pour la recherche d'information.

**Mots-clés** : Visualisation, interaction, recherche d'information.

## Abstract

Information retrieval is no longer reserved to professional librarians. It has become a common activity for a large number of users. User interfaces to information retrieval systems are often very primitive, consisting only of a text field to enter a query and a button to validate it. Many innovative visualisation and interaction techniques have been developed over the last ten years in order to address the challenge of making information retrieval accessible to a wide variety of users and uses. Unfortunately these techniques are rarely used in actual systems. This articles summarizes the current state of the art in visualization and interaction that is relevant to the field of information retrieval.

**Keywords**: Visualization, interaction, information retrieval.

## INTRODUCTION

Si les moteurs de recherche jouent un rôle prépondérant pour la recherche d'information leurs interfaces graphiques sont rarement révolutionnaires : un champ texte uniligne pour la saisie de la requête, un bouton pour la valider, et un affichage des résultats sous la forme de listes parfois arborescentes. La navigation dans la liste se fait en ayant recours à des mécanismes de défilement (barre de défilement ou molette de la souris) et à la métaphore de la page pour un défilement à plus gros grain. Une alternative couramment proposée à cette approche orientée requête consiste à proposer un catalogue arborescent de thèmes. Mais, là encore, les modes de visualisation ou d'interaction relèvent de modèles primaires. La plupart des catalogues sont représentés sous forme de tableaux arborescents, et la navigation hypertexte permet de descendre dans la hiérarchie de thèmes proposés.

Que la majorité des interfaces graphiques utilisées actuellement pour la recherche d'information en soit encore à l'âge de pierre n'est pas une fatalité. Il est bien normal que jusqu'à maintenant les efforts de développement se soient portés essentiellement sur les moteurs de recherche, clés de voûte du système. D'une part, le domaine de la recherche d'information plein-texte est beaucoup plus mature que celui de la visualisation, d'autre part, les ressources mises en œuvre pour faire tourner un moteur de recherche (en terme d'architectures, de machines, de mémoires, etc.) sont contrôlées totalement par le producteur du moteur et peuvent être augmentées quasiment à l'infini pour faire face à une demande de puissance considérable. Le problème de l'interface graphique est tout autre. En effet, l'interface graphique ne dispose généralement pour tourner que des ressources propres à la machine sur laquelle elle s'exécute. Dans le contexte du Web par exemple, le concepteur de système de recherche d'information ignore tout ou presque de cette machine cliente et distante. Par conséquent, la plupart des interfaces graphiques sont conçues pour exploiter un minimum de ressources, et les tentatives de construction d'interfaces graphiques avancées ont été jusqu'à maintenant prématurées dans un environnement où ce minimum était insuffisant. Ainsi, les tentatives d'Apple avec

HotSauce pour visualiser les termes d'une requête ou les pseudo-vues offertes par Altavista ont rapidement été abandonnées. Ce sont souvent les ressources disponibles qui conditionnent le succès d'une interface graphique avancée, car les temps de réponse ou la richesse du graphique qui en dépendent sont des éléments incontournables de ce succès.

Néanmoins, l'évolution des postes de travail individuels et l'essor considérable qu'a connu le domaine de la visualisation interactive d'information depuis plus de dix ans permettront à plus ou moins court terme de passer à l'étape suivante. On peut alors s'attendre à une mutation aussi importante que celle liée au passage de la visualisation alphanumérique à celui de la manipulation directe. Témoin de cette mutation, le méta-moteur Kartoo (Baleyrier, 2001) est parmi l'un des premiers méta-moteurs publics à réussir un changement radical d'interface graphique (cf. Fig. 1). Comme tout méta-moteur Kartoo bénéficie de techniques de recherches avancées. Le travail du méta-moteur se fait en amont et en aval d'autres moteurs. En amont, il s'agit de traduire des opérateurs avancés pour différents moteurs ou encore de développer des heuristiques de redirection de requêtes. Dans Kartoo, les requêtes courtes sont redirigées vers Google exploitant ainsi, entre autres, la notion de "hub" et "d'authority" mises en œuvre dans ce moteur (Kleinberg, 1998). Les requêtes plus longues sont au contraire redirigées vers NorthernLight pour exploiter l'aspect analyse du langage naturel.



FIG. 1 – Kartoo, un méta-moteur de recherche qui combine technique de fouille de texte et nouvelles stratégies de présentation.

En aval, il s'agit d'analyser les résultats de façon à simplifier la tâche de l'utilisateur. Outre les analyses de base, permettant par exemple le filtrage des sites jumeaux, Kartoo utilise des techniques de fouille de données textuelles (« text mining ») pour analyser le texte recueilli de sorte à déterminer statistiquement, et bientôt sémantiquement, les principaux thèmes abordés dans les documents résultant. C'est alors que Kartoo quitte les sentiers battus en construisant une représentation graphique des résultats. Cette vue de synthèse (Fig. 1) est composée de boules représentant les sites pertinents et le diamètre de ces boules reflète la pertinence du site pour la requête. Les thèmes extraits automatiquement par Kartoo sont également affichés et liés aux sites d'où ils sont extraits. Certes Kartoo n'est pas la première initiative dans ce domaine : construire des cartes pour le Web a été l'objet de nombreux travaux (Kahn, 1999), mais l'originalité de Kartoo est de proposer une vue minimale mais parfaitement bien conçue. A partir de cette vue, l'utilisateur peut choisir d'explorer rapidement les sites pertinents ou d'entrer dans un processus interactif de raffinement de requête. Dans ce processus, le contrôle des termes clés ajoutés ou supprimés se fait de manière coopérative : le méta-moteur les extrait, l'utilisateur les exploite. Ainsi, en agissant sur les thèmes affichés graphiquement l'utilisateur peut-il reformuler sa requête de manière adéquate et en voir les effets graphiquement. Cette vue, combinée aux techniques de recherche avancées qui permettent de la construire, et à l'exploitation par manipulation presque directe, change radicalement la manière dont les résultats d'une requête peuvent être appréhendés et exploités. Cette initiative n'est pas isolée, et bien d'autres progrès sont à attendre de la combinaison des récentes avancées en terme de recherche d'information, de visualisation et d'interaction.

Dans cet article nous présentons une synthèse des travaux de recherche de ces dix dernières années en matière de visualisation et d'interaction dans le contexte de la recherche d'information. Nous supposons que le lecteur est familier avec les techniques de base de la recherche d'information plein-texte (modèle vectoriel, calcul du coefficient de pertinence, calcul de similarité, injection de retour de pertinence, reformulation de requête, etc.), qui sont décrites dans des ouvrages tels que Salton et McGill (1983), Salton (1989), Witten et al. (1999). Dans la première partie de l'article, nous abordons les problèmes de visualisation de listes, d'arbres, de

graphes et d'espaces munis d'un indice de similarité. Cette taxonomie, bien que très sommaire permet de donner un aperçu global des stratégies de présentation de l'information pertinentes dans le cadre de la recherche plein-texte. Nous n'aborderons pas les problèmes liés à la construction d'une taxonomie plus élaborée qui sont traités par exemple dans Keim et Kriegel (1996), Shneiderman (1996), Card et Mackinlay (1997), Tweedie (1997), Bruley et Genoud (1998). La plupart des stratégies nouvelles de présentation de l'information requièrent de nouveaux modèles d'interaction. Ces modèles font l'objet de la deuxième partie. Nous présentons différentes approches incluant le filtrage dynamique, l'intégration de détails dans leur contexte ou encore le zoom infini.

## VISUALISATION

Le but de la visualisation d'information est d'exploiter les caractéristiques du système visuel humain pour faciliter la manipulation et l'interprétation de données informatiques variées. Les travaux en perception visuelle ont montré que l'être humain a une perception d'abord globale (gestalt-perception) d'une scène, avant de porter son attention aux détails (Myers, 2000). Cette caractéristique est à la base de nombreuses illusions visuelles. Les travaux de Tufte (1983) et Bertin (1977) ont montré comment exploiter, de façon intuitive ou ad hoc, ces caractéristiques de perception globale. La visualisation d'information cherche à exploiter ces mêmes caractéristiques de façon plus systématique. Il faut cependant noter que l'efficacité d'une technique de visualisation est difficile voire impossible à évaluer de façon absolue. Tout au plus peut-on espérer l'évaluer par rapport à un contexte particulier : tâche de l'utilisateur et caractéristiques des données.

La visualisation d'information étant destinée à explorer des données, les tâches rencontrées en visualisation d'information sont liées à la recherche d'information au sens large :

- exploration rapide d'ensembles d'informations inconnues ;
- mise en évidence de relations et de structures dans les informations ;
- mise en évidence de chemins d'accès à des informations pertinentes ;
- classification interactive des informations.

La plupart des informations que l'on peut souhaiter visualiser dans ce contexte peuvent être classées en quatre grandes catégories selon leur « nature ». Ces quatre catégories correspondent à des ensembles de données dont les propriétés diffèrent. La première catégorie correspond à des ensembles qui peuvent être vus comme des listes. La deuxième catégorie correspond à des ensembles qui peuvent être structurés de manière arborescente. La troisième catégorie comprend les ensembles dont on peut extraire des structures de graphe. Enfin, la quatrième catégorie correspond aux ensembles pour lesquels on peut exploiter un indice de similarité (Salton et McGill 1983).

Par exemple, les documents résultant d'une requête et classés par ordre de pertinence entrent dans la première catégorie. Des documents correspondant aux différentes parties d'un livre – chapitres, sections, paragraphes ou bien encore des pages WWW classées dans une liste de « favoris » sont des ensembles de documents qui appartiennent à la deuxième catégorie. Des pages WWW issues d'un site quelconque et liées entre elles par des liens hypertextes appartiennent à la troisième catégorie. Enfin, selon la technique de recherche employée, les documents issus d'une requête peuvent être comparés par une mesure de similarité et correspondre alors à la quatrième catégorie. De manière générale, le même ensemble de documents peut subir des traitements afin de faire apparaître l'une ou l'autre des propriétés qui caractérisent chacune de ces catégories. Ainsi le même ensemble de documents pourra potentiellement appartenir à plusieurs catégories, mais la technique de visualisation que l'on utilisera dépendra des propriétés que l'on privilégie.

### Visualisation de listes

Les techniques couramment utilisées pour visualiser des listes sont une page ou une suite de pages avec des barres de défilement. Cependant ces techniques sont peu adaptées à des listes de grande taille. D'une part, la surface occupée pour les présenter est importante. D'autre part, le parcours de la liste est souvent lent. Enfin, la proportion des documents visibles par rapport aux documents cachés est faible dès que la liste devient grande.

*Document Lens* (Robertson et Mackinlay, 1993) et le *mur fuyant* (Mackinlay et al., 1991) sont des techniques de visualisation de listes qui se basent sur une transformation géométrique pour déformer une grille ou une liste et relèvent d'une approche générale visant à concilier détails et contexte dans une même vue. D'autres approches se contentent d'augmenter les techniques actuelles par des indices visuels qui résument diverses propriétés des documents : *TileBars* (Hearst, 1995), *SeeSoft* (Eick et al., 1992), *Value Bars* (Chimera, 1992).

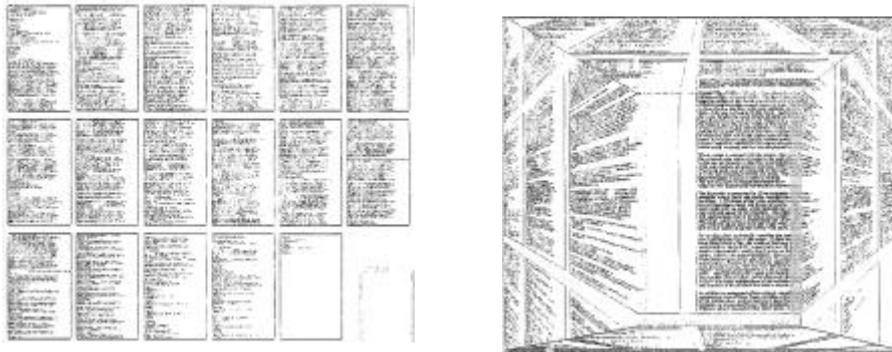


FIG. 2 – Document lens: visualisation des documents sur une grille (à gauche) et déformation pour obtenir plus de détails sur certains éléments particuliers tout en gardant le contexte

### Déformations pour l'intégration de détails dans un contexte

Dans le système *Document Lens* (Robertson et Mackinlay, 1993) (cf. Fig. 2), les documents sont affichés sous la forme d'images de pages et sont disposés sur une grille dans l'ordre de la lecture. Ainsi dans la vue d'ensemble les différents documents ne sont pas lisibles mais leur aspect visuel miniature constitue un indice qui peut permettre de se déplacer directement vers le document recherché. Lorsque l'utilisateur pointe vers l'un des documents de la grille, une déformation de la grille permet de rendre ledit document lisible, tout en gardant le contexte des autres documents. Avec cette technique, le parcours de la liste de documents peut être plus rapide, car l'affichage du contexte aide à l'orientation et aux déplacements directs. La proportion du visible par rapport à l'invisible est également améliorée car un grand nombre d'informations peut être visualisé simultanément. Par contre, la surface nécessaire à l'affichage de ces présentations reste particulièrement importante.

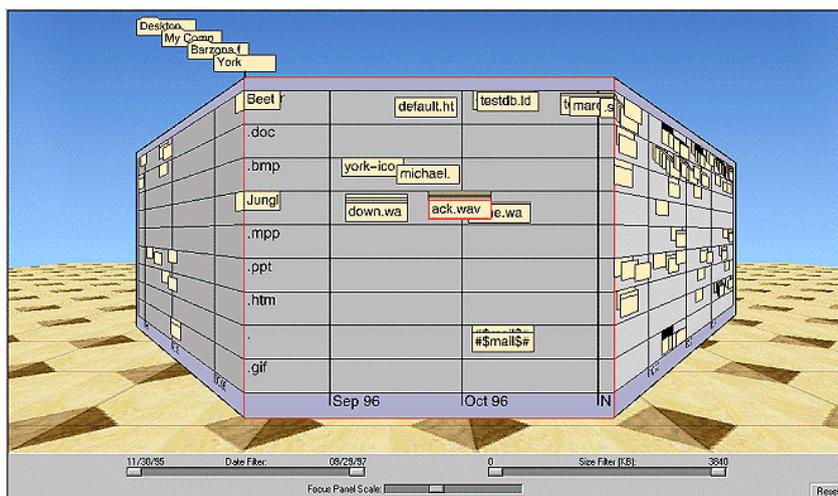
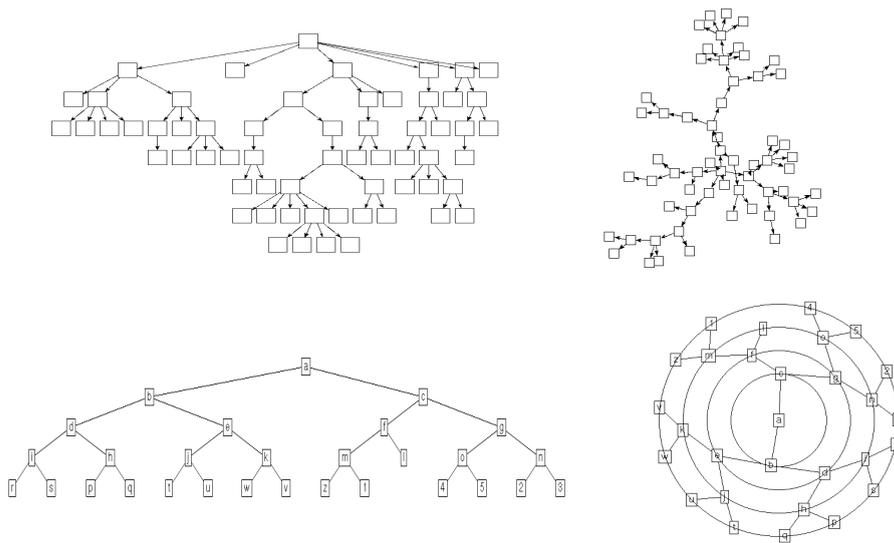


FIG. 3 – Mur Fuyant

Comme d'autres systèmes, *Document Lens* utilise le greeking pour afficher le texte en petite taille. Cette technique consiste à représenter chaque ligne par un segment de droite de longueur proportionnelle au nombre de caractères de la ligne. Le recours au greeking est une pratique courante en visualisation d'information car elle permet de contourner les problèmes liés à l'affichage du texte tout en donnant des indices permettant de reconnaître par leur forme les passages caractéristiques.

Le *mur fuyant* (Mackinlay et al., 1991) (cf. Fig. 3) procède de la même démarche en conciliant détails et contexte grâce à une déformation géométrique mais la représentation des informations de départ est différente : les informations sont représentées par des étiquettes placées sur un mur en utilisant un axe horizontal et vertical. L'axe horizontal représente l'ordre dans la liste et l'axe vertical permet de représenter un autre attribut des données. Une déformation de type bifocale (Leung et Apperley, 1994) est utilisée pour faire apparaître plus de détails sur la partie centrale du mur tandis que le reste du mur apparaît en perspective. L'utilisateur peut contrôler le niveau de détail de la partie centrale du mur, avec un effet qui ressemble à l'étirement d'une surface élastique.





**FIG. 6** – diagramme d'arbres algorithme de Walker (gauche haut), binaire (gauche bas), algorithme général de placement de graphe de Kamada (droite haut), radial (droite bas).

## Visualisation d'arbres

Généralement, des informations organisées hiérarchiquement (catalogues thématiques de sujets, documents organisés en chapitres, sections, etc.) sont visualisées sous forme de listes indentées. Ce type de représentation comporte un inconvénient majeur pour des arborescences de grande taille car elle ne parvient pas à donner une vue d'ensemble satisfaisante : l'arbre n'est jamais entièrement visible car le nombre de nœuds affichés simultanément est forcément limité par l'espace disponible. Comme pour les listes traditionnelles cela rend inefficace un certain nombre de tâches liées à la manipulation/exploitation de l'arbre.

D'autres techniques ont été développées pour les arbres de plus grande taille. Ces techniques privilégient la visualisation de l'ensemble de l'arbre aux dépens des détails de chaque élément. Ces détails pourront être obtenus par interaction sur l'arbre. On peut distinguer quatre approches de visualisation très différentes :

- Approche diagramme (Battista et al., 1999; Herman et al., 2000)
- Approche "surfactive" : cartes d'arbres (Shneiderman, 1992), arbres circulaires (Andrews et Heidegger, 1998)
- Approche 3D : les arbres coniques (Robertson et al., 1991)
- Approche géométrie non-euclidienne : les arbres hyperboliques (Lamping et al., 1995)

### Approche diagramme

Une première alternative à l'approche usuelle de listes arborescentes consiste à représenter l'arbre par un diagramme constitué de nœuds et de branches. Dans cette approche, le principal problème consiste à définir l'algorithme idéal de placement d'arbre. Pour cela, de nombreux algorithmes existent, surtout si l'on considère qu'un arbre étant un cas particulier de graphe, tous les algorithmes permettant de dessiner un graphe peuvent s'appliquer (Battista et al., 1999; Herman et al., 2000).

Parmi les algorithmes les plus couramment utilisés dans le cadre de la visualisation d'information, on peut distinguer ceux pour lesquels les caractéristiques principales conduisent à mettre en valeur l'arborescence en utilisant une représentation usuelle horizontale ou verticale (cf. Fig. 6) et ceux qui visent à mieux tirer parti de l'espace disponible en adoptant une représentation concentrique.

Dans le premier cas, les algorithmes de placement cherchent à satisfaire un certain nombre de contraintes : centrer un nœud au-dessus de ses enfants, respecter un espace inter-nœuds ou dessiner de façon similaire des sous-arbres isomorphes. Dans certains cas, les algorithmes permettent d'optimiser la largeur du graphe. Parmi les algorithmes utilisés dans la Fig. 6 pour produire un dessin d'arbre vertical, le premier est une généralisation d'un algorithme de placement binaire et présente l'avantage d'être extrêmement simple à

implémenter (Battista et al., 1999). Le deuxième (Walker, 1990) est moins simple mais donne de meilleurs résultats en minimisant entre autre la largeur utile pour le dessin.

Les algorithmes du deuxième type partagent une partie de ces contraintes mais intègrent une contrainte supplémentaire : tirer le meilleur parti de l'espace disponible. A cet effet, les algorithmes radiaux utilisent mieux l'espace que les représentations verticales ou horizontales : les nœuds de chaque niveau de l'arbre sont répartis sur des cercles ou ellipses concentriques dont le rayon augmente avec la profondeur dans l'arbre. Ce type de représentation est particulièrement bien adapté à des arbres relativement peu profonds pour lesquels il existe un grand nombre de feuilles (comme par exemple certains sites Web ou certains catalogues de sites). Nous avons remarqué deux effets contrastés avec l'utilisation de ce type de représentations: l'arborescence est moins bien perçue par une majorité d'utilisateurs, et pourtant ce type de vue est en général préféré (Hascoët, 2001).

L'utilisation des diagrammes ne résout pas tous les problèmes liés au volume de l'information à visualiser. Ainsi l'approche sous forme de diagramme peut-elle être envisagée avec succès pour des arbres contenant plusieurs centaines de nœuds mais elle l'est rarement pour des arbres de plusieurs milliers de nœuds.

### **Approche surfacique**

Les cartes d'arbres ou « tree-maps » (Shneiderman, 1992) ont justement pour but de traiter des arbres de grande taille. Leur principe consiste à visualiser un arbre en découpant une surface donnée (en général rectangulaire) proportionnellement à chaque sous-arbre. Chaque rectangle représente une feuille de l'arbre et sa surface est proportionnelle à son importance dans l'arbre. De même chaque regroupement horizontal ou vertical de rectangles représente un sous-arbre proportionnellement à son importance (cf. Fig. 7). Cette approche permet de visualiser rapidement l'ensemble de l'arbre pour déceler certains éléments particuliers, la couleur ainsi que le motif de fond peuvent être utilisés pour représenter certaines caractéristiques des nœuds.

Les arbres représentés par des demi-disques (cf. Fig. 8) relèvent d'une démarche analogue (Andrews et Heidegger, 1998). Chaque niveau de l'arbre est représenté par un anneau que les nœuds se partagent. Par rapport aux arbres planaires, l'avantage de cette représentation est qu'elle rend l'arborescence beaucoup plus facilement perceptible. Par contre, l'affichage de texte à l'intérieur d'un nœud est quasiment impossible car les nœuds ont la forme de secteurs d'anneaux. A vrai dire ce problème se manifeste également avec les tree-maps lorsque les rectangles sont très étirés. Cependant, il existe des variantes des tree-maps permettant d'optimiser les proportions des rectangles représentant les nœuds de façon à pouvoir afficher des informations textuelles dans ces rectangles.

### **Dessin d'arbres en 3 dimensions**

Une autre approche consiste à avoir recours à la 3D pour visualiser un plus grand nombre de nœuds (cf. Fig. 9). Les arbres coniques sont le meilleur exemple de cette approche (Robertson et al., 1991; Card et al. 1991). Ils permettent d'afficher des arbres de grande taille de façon intuitive. L'inconvénient majeur de ces techniques est lié à l'usage de la 3D et notamment aux problèmes d'occlusion, de manipulation et d'orientation qui lui sont inhérents.

### **Géométrie hyperbolique**

Le recours à la géométrie hyperbolique (Lamping et al. 1995) pour représenter des arbres (cf. Fig. 10) peut être vu comme un cas particulier de l'approche diagramme. Les caractéristiques très particulières des transformations de la géométrie hyperboliques justifient néanmoins qu'on l'en distingue. Selon cette approche les nœuds de l'arbre sont placés de manière analogue à l'approche radiale sur des cercles concentriques dont le rayon est calculé proportionnellement à la profondeur du nœud dans l'arbre. L'intérêt de la géométrie hyperbolique est qu'ici le périmètre des cercles hyperboliques croît exponentiellement avec leur rayon ce qui permet de visualiser facilement des arbres contenant un grand nombre de feuilles.

Les transformations de la géométrie hyperbolique permettent d'interagir avec la vue et en particulier de déplacer des nœuds d'intérêts au centre. Ces transformations très particulières ont un effet un peu surprenant pour un utilisateur non entraîné, comparable à l'effet obtenu lors d'un placement radial avec application d'une transformation fisheye (cf. Fig. 17 par exemple). Dans les deux cas, les utilisateurs non entraînés ont besoin de faire un effort cognitif non négligeable pour reconstituer l'effet des transformations sur le contexte qui se déplace de manière inhabituelle. Il reste à savoir si cette géométrie et les transformations qui en résultent peuvent être maîtrisées après quelques séances pour permettre d'exploiter pleinement le bénéfice de cette technique.

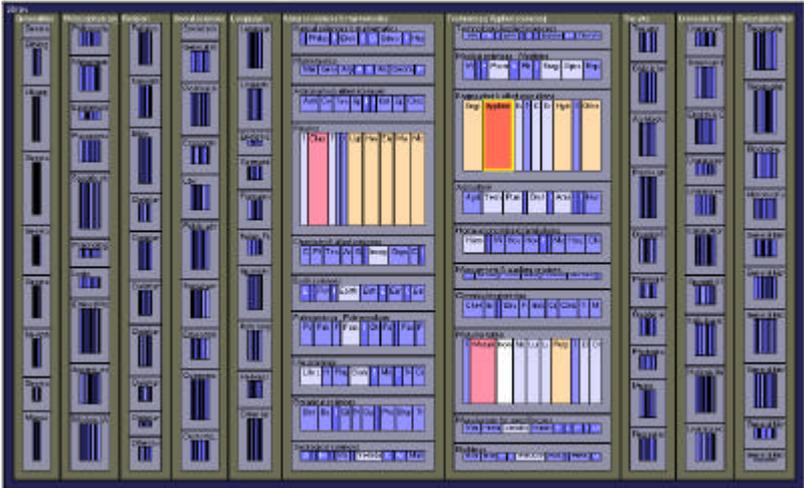


FIG. 7 – Tree-maps

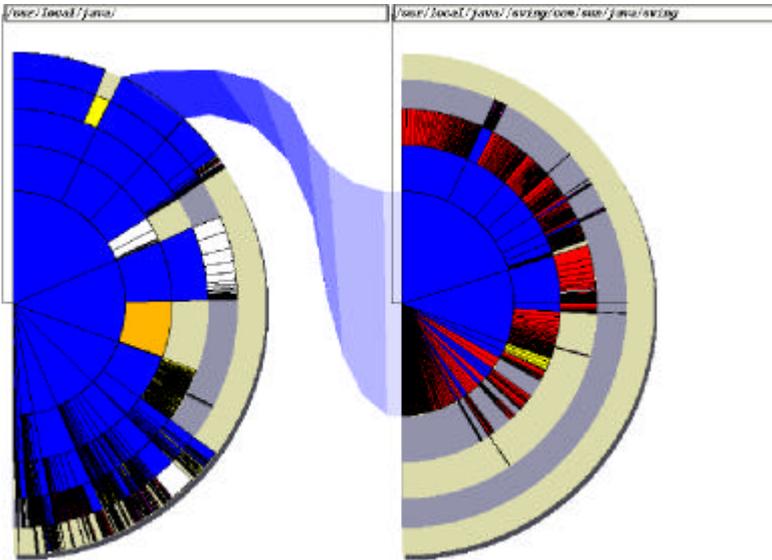


FIG. 8 – Disques d'arbres

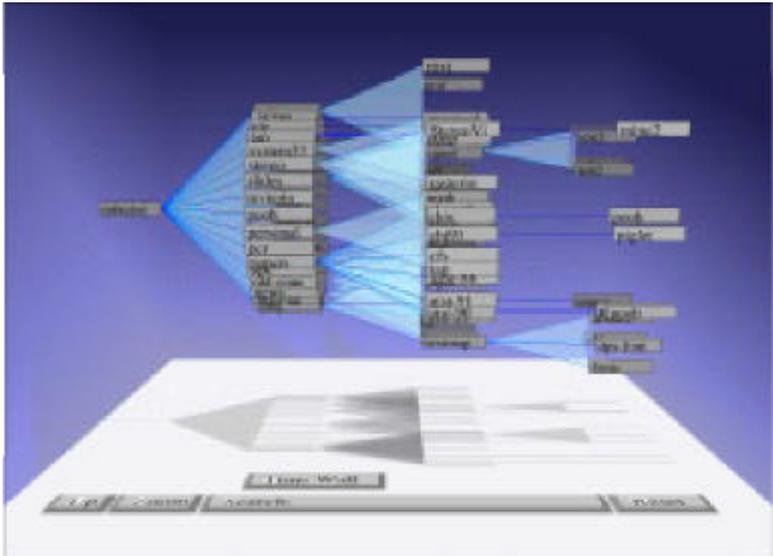


FIG. 9 – Arbres coniques



Ces algorithmes sont difficilement utilisables pour visualiser des graphes de grande taille car ils sont particulièrement coûteux en temps de calcul, leur complexité étant en général au mieux polynomiale mais le plus souvent exponentielle (par rapport au nombre de nœuds). D'autres algorithmes existent et un état de l'art sur la visualisation de graphe dans le contexte de la visualisation d'information pourra être trouvé dans Battista et al. (1999) ou Herman et al. (2000).

## Visualisation pour le modèle vectoriel

Les modèles conventionnels de document issus de la recherche d'information consistent à représenter des documents par des vecteurs (sortes d'index). Les coordonnées de ces vecteurs dépendent du modèle utilisé (Salton et McGill, 1983). Généralement, les coordonnées sont calculées par rapport à une base de mots-clés et reflètent la présence ou l'absence de ces mots-clés dans un document.

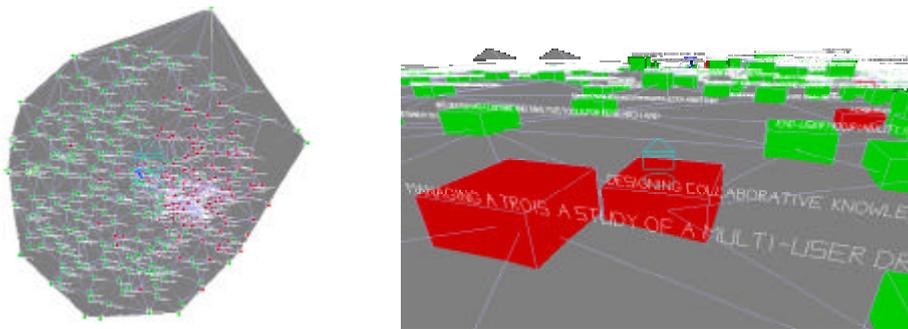
Des procédés de calcul issus de ce modèle permettent de définir des mesures de similarités ou de pertinence entre différents « morceaux » d'information (documents, requêtes, thème, etc.). L'utilisation de ces structures de représentation a conduit au développement de techniques de visualisation spécifiques exploitant les caractéristiques de ces structures, en particulier pour les calculs de similarité.

### Métaphore du paysage

Parmi ces techniques de visualisation, on trouve celles développées avec la métaphore du paysage (Chalmers, 1993; Chalmers et Chitson, 1992), où des documents sont placés dans un paysage virtuel en trois dimensions (cf. Fig. 11). Dans ce paysage, les documents similaires – au sens de la mesure de similarité – sont placés de telle sorte qu'ils soient voisins. Les algorithmes qui permettent ce type d'approche sont décrits dans Chalmers (1993), Chalmers et Chitson (1992).

### Utilisation d'un espace distance

D'autres techniques de visualisation pour ce type de structures ont été développées dans le système *VIBE* (Olsen et al., 1993) (cf. Fig. 12, gauche). Leur principe consiste à utiliser les points d'un espace 2D pour représenter les informations (par exemple des documents), ces points sont placés dans l'espace en fonction de points de référence (représentant en général des mots-clés). Le placement de ces points est calculé de telle sorte que la distance euclidienne des points-documents aux points-références illustre la pertinence des documents par rapport aux mots-clés. Ce principe de placement a été repris en 3D dans le système *Lyberworld* développé au GMD (Hemmje et al., 1994) (cf. Fig. 12, droite).



**FIG. 11** – Métaphore du paysage - à partir de la vue d'ensemble (à gauche) l'utilisateur peut naviguer en 3D pour se rapprocher des documents qu'il cherche (à droite).

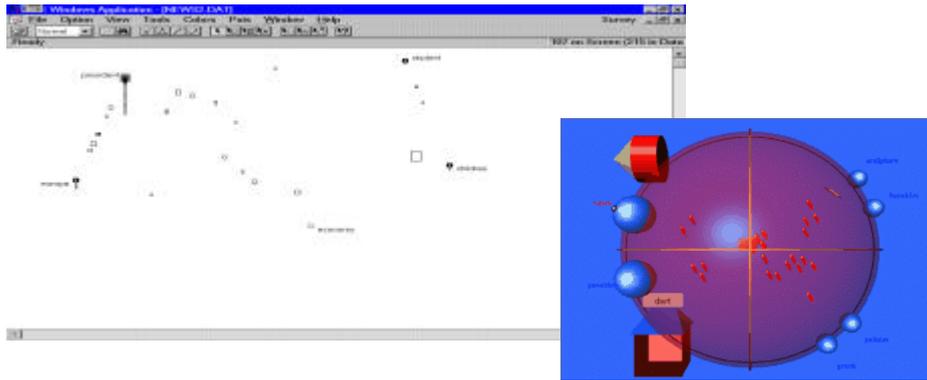


FIG. 12 – Visualisation dans un espace distance en 2D (à gauche) et en 3D (à droite)

### Cartes interactives dynamiques

Enfin, un autre type de visualisation pour ce type de structures a été développé dans les *Cartes Interactives Dynamiques* (Zizi et Beaudouin-Lafon, 1995), où des informations (documents, mots-clés, thèmes, etc.) sont placés sur des cartes de telle sorte que les informations les plus similaires soient voisines (cf. Fig. 14). Plusieurs types de cartes ont été développées pour des contextes divers. Les cartes de thesaurus par exemple sont des cartes obtenues par analyse plein-texte d'une collection de documents. Cette analyse permet l'extraction automatique de terminologie et les termes extraits sont ensuite soumis à une classification automatique. Il en résulte un thesaurus primaire qui est visualisé sous forme de carte. Les cartes de documents constituent un autre type de cartes dans lesquelles des documents sont représentés de telle sorte que les documents proches au sens de la mesure de similarité soient voisins sur la carte.

## INTERACTION

Pour être réellement efficace, la visualisation d'information doit être interactive. En effet, l'être humain est particulièrement habile à extraire des informations d'un environnement qu'il contrôle directement et activement par rapport à un environnement qu'il ne peut qu'observer de manière passive. Selon l'approche écologique de la perception due au psychologue J.J. Gibson (1979), la perception est indissociable de l'action : il faut agir pour percevoir et il faut percevoir pour agir. On parle de couplage (ou boucle) action-perception. De plus, la perception de notre environnement consiste à extraire des flux perçus (comme le flux visuel) des *invariants*. Par exemple, lorsque l'on se déplace, la direction du déplacement est donnée par le seul point immobile dans le flux visuel. Grâce à l'interaction sur les données, l'utilisateur peut agir sur ce qu'il perçoit et, par l'extraction d'invariants, mieux comprendre la nature des données ou de leur processus de représentation.

La visualisation interactive requiert des temps de réponse interactifs (de l'ordre du 1/10<sup>ème</sup> de seconde) afin d'exploiter le couplage action/perception. Dès que l'on traite des données de grande taille, cette contrainte exige l'utilisation d'algorithmes particulièrement efficaces. Pour les besoins de l'interaction, il est parfois préférable de dégrader la qualité de l'affichage au profit de la vitesse de réaction, quitte à raffiner l'affichage lorsque l'interaction s'arrête ou marque une pause. Ainsi, au couplage action-perception de l'utilisateur correspond un fort couplage affichage-interaction du système de visualisation.

Un certain nombre de techniques classiques d'interaction peuvent être utilisées pour la visualisation d'information, comme par exemple les barres de défilement pour naviguer le contenu d'une fenêtre, ou les boîtes de dialogues pour spécifier des paramètres de visualisation. Cependant, les meilleurs résultats sont obtenus en couplant plus fortement l'interaction avec la technique de visualisation. Trois catégories de techniques d'interaction peuvent être mises en œuvre avec un grand nombre de techniques de visualisation : (1) le filtrage dynamique, (2) les transformations sémantico-géométriques et (3) les zooms spécifiques.

### Filtrage dynamique

Le filtrage dynamique est particulièrement utile pour interagir avec des vues globales, car l'information présente dans ces vues n'est pas définitivement pertinente pour un utilisateur. En effet, s'il est intéressant pour un utilisateur de voir dans un premier temps l'ensemble des informations qu'il explore pour mieux

l'appréhender, il sera intéressant par la suite que l'utilisateur puisse recentrer son exploration, en réduisant son espace de recherche.

Les requêtes dynamiques (Ahlberg, et al., 1992; Williamson et Shneiderman, 1992; Ahlberg et Wistrand, 1995; Ahlberg et Shneiderman, 1994) développées à l'Université de Maryland puis à IVEE (système SpotFire) constituent un bon exemple de mode d'interaction permettant ce type de filtrage dynamique. Leur principe repose sur un fort couplage entre d'une part des éléments d'interaction (curseurs, menus de sélection), permettant de spécifier graphiquement des requêtes et d'autre part la vue globale sur les données (cf. Fig. 13) qui se remet à jour immédiatement en fonction de la requête spécifiée. Ainsi, l'activation d'un curseur sur le panneau de contrôle permet de voir instantanément l'effet du changement de valeur correspondant sur la vue globale. Des données deviennent dynamiquement visibles ou invisibles sous l'action de l'utilisateur. Cette technique permet à l'utilisateur de réaliser le filtrage en fonction du résultat qu'il obtient et non pas en fonction du contrôle qu'il opère. Par exemple, il peut aisément détecter des situations où un faible changement d'une valeur de paramètre produit un effet important sur les données filtrées, et explorer plus en détail ces zones. Le succès de ce type de technique est fortement dépendant des temps de réponse du système.

Un autre type de filtrage dynamique a été développé par Furnas (1986) Il s'agit d'une technique appelée *fisheye*. Cette technique a été appliquée à des visualisations d'arbres et permet de faire apparaître ou disparaître des nœuds de l'arbre en fonction de leur degré d'intérêt (Degree Of Interest ou DOI). Le DOI dépend (1) de l'importance a priori des nœuds (IAP) et (2) de leur distance au nœud pointé par l'utilisateur ou focus. L'importance a priori des nœuds est donnée, elle dépend des informations visualisées, tandis que la distance entre un nœud et le focus est calculée comme le plus court chemin entre les deux.

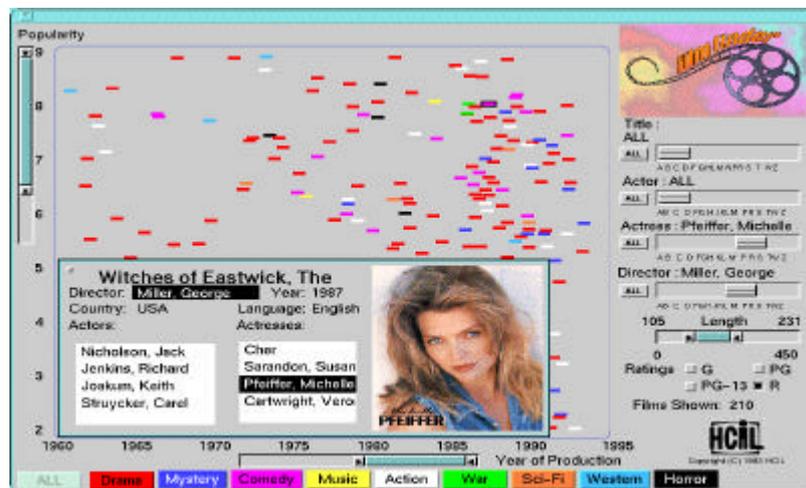


FIG. 13 – Starfield

Il est important de reconnaître que les techniques de fisheye existent dans deux variantes très différentes du point de vue de leur utilisation. En effet, la technique de fisheye « filtrant » telle qu'elle a été proposée par Furnas et décrite au paragraphe précédent, diffère des techniques de fisheye « déformants » proposés par exemple par Sarkar et Brown (1992) et décrites plus loin. Dans le premier cas, il s'agit bien de filtrage dynamique puisque des informations vont devenir visibles ou invisibles sous l'action de l'utilisateur, dans le deuxième cas il s'agit d'un mode d'interaction permettant de concilier détails et contexte puisque aucune information ne disparaît ou n'apparaît, les informations affichées sont seulement déformées. Les deux approches partagent néanmoins un concept commun : le calcul du degré d'intérêt (DOI) qui tient compte du centre d'intérêt courant de l'utilisateur pour effectuer le filtrage dans un cas et les déformations dans l'autre.

Enfin, notre approche du filtrage dynamique (Hascoët-Zizi et Pediotakis 1996) a consisté à coupler une carte de documents avec une carte de thesaurus (cf. cartes interactives dynamiques) et à interagir avec la carte de thesaurus pour filtrer les documents de la carte de documents (cf. Fig. 14) : la sélection de termes et de relations dans la carte de thesaurus spécifie une requête dont le résultat s'affiche dans la carte de documents.

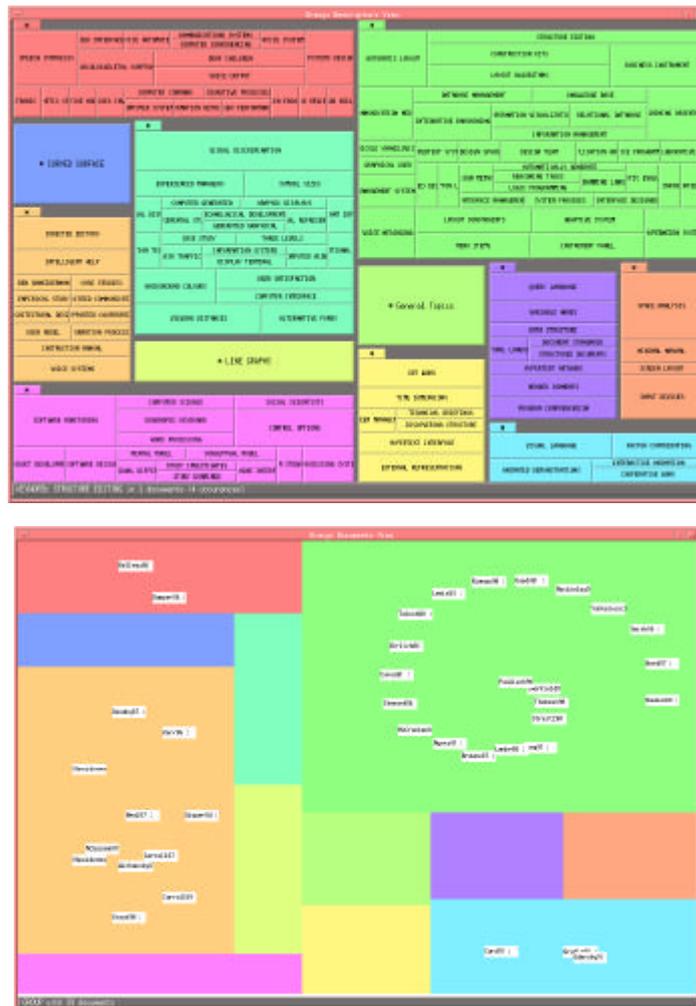


FIG. 14 – Analyse visuelle de la pertinence par filtrage dynamique de la vue de documents (en bas) par la vue du thesaurus associé (en haut)

## Integration de différents niveaux de détails

Le filtrage déformant a pour effet de rendre visible ou invisible des données. Souvent, ces changements brusques de visibilité rendent l'exploration difficile. La notion de niveau de détail permet une approche plus graduée, en permettant d'amplifier dynamiquement les détails autour d'un centre d'intérêt tout en gardant le contexte global de la vue. Dans ce domaine, de nombreuses solutions ont été proposées. Nous présenterons ici les fisheye déformant de graphe (Sarkar et Brown, 1992) et fisheye déformant structurels (Noik, 1993a; Noik 1993b; Bartram et al., 1995; Schaffer 1996). De nombreuses autres techniques visent le même objectif : *Polyfocal display*, *Bifocal display* ou même *Mur Fuyant* (Mackinlay et al., 1991) présenté plus haut. Pour une revue plus complète de ces techniques de déformation, le lecteur peut se référer à Leung et Apperley (1994). Une approche unificatrice de ces techniques, appelée *Scrolling Généralisé*, est présentée dans Smith et Taivalsaari (1999).

La transformation fisheye proposée par Sarkar et Brown opère sur un graphe valué. Les poids des nœuds constituent leur importance a priori IAP. La transformation fisheye consiste à modifier les positions et les tailles des nœuds affichés à l'écran (cf. Fig. 16). Les nouvelles tailles sont obtenues en fonction du degré d'intérêt DOI calculé pour chaque nœud. Les nouvelles positions sont obtenues en appliquant une fonction de déformation : si O est le nœud focus, et M est un nœud à transformer, le calcul de k tel que  $OM = k OS$  (S est l'intersection de (OM) avec les frontières de la zone d'affichage) permet de calculer l'image de M. Cette image M' est telle que  $OM' = f(k) OS$ , avec  $f: [0,1] \rightarrow [0,1]$  et  $f(k) > k$ , et f' décroissante sur  $[0,1]$ . Ainsi les nœuds les plus proches du focus subissent un déplacement plus important que les nœuds plus éloignés.

## Fisheye déformant

Il existe plusieurs variantes de cette formulation de la transformation (voir par exemple Sarkar et Brown, 1992). La Fig. 17 montre l'effet de deux formulations différentes : l'une globale calculée en coordonnées cartésiennes et l'autre locale à une loupe calculée en coordonnées polaires. Le résultat de cette double déformation (modification des tailles et modification des positions) a ainsi pour effet d'amplifier les nœuds se trouvant dans la région du focus tout en repoussant et en diminuant l'importance des nœuds situés en dehors de cette région (cf. Fig. 17). Comme avec l'utilisation de la géométrie hyperbolique, l'effet de déformation peut-être perturbant pour l'utilisateur, surtout si la distance et/ou la disposition des données a une signification particulière (comme pour des données géographiques). Une alternative consiste à utiliser une vue séparée pour représenter les détails, comme avec une loupe mais alors la vue détaillée masque le contexte. Le *DragMag* (Ware et Lewis, 1995) évite cet effet de masquage en décalant la vue magnifiée. Dans tous les cas, c'est la qualité et la rapidité du couplage entre l'action sur la vue et son effet sur la représentation qui sont la clé de ces techniques.

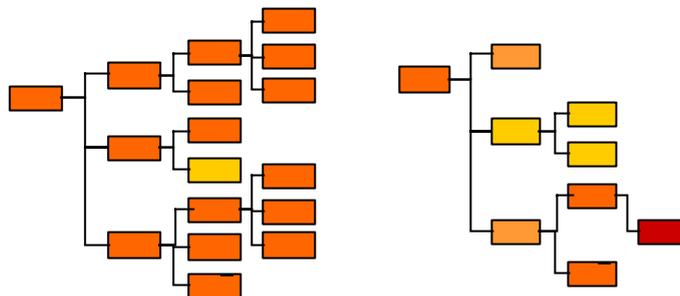


FIG. 15 – Fisheye filtrant : à gauche l'arbre original, à droite l'arbre après fisheye.

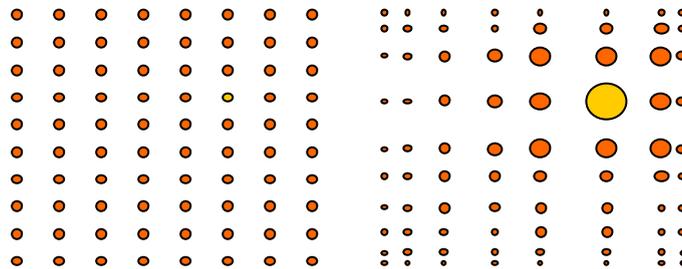


FIG. 16 – Fisheye déformant : le résultat (à droite) est obtenu à partir d'une vue initiale (à gauche) et d'un centre d'intérêt (en plus gros).

## Fisheye déformant et structurel

La technique de transformation fisheye utilisée par Dill et testée par Schaffer (1996) est reposée sur la transformation fisheye d'un graphe simple telle que celle proposée par Sarkar et Brown, mais elle va plus loin car elle profite de l'espace gagné sur certains nœuds pour les "développer". En effet, le fisheye déformant et structurel s'applique récursivement à un graphe organisé en nœud-nuées. Les nœuds-nuées peuvent être vus comme des nœuds composés qui représentent un ensemble de nœuds sur lesquels les déformations doivent être répercutées. Ainsi, le fisheye (Noik, 1993) s'applique-t-il récursivement pour que les nœuds-nuées grossissent lors de la transformation répercutent ce grossissement sur les nœuds qui les composent. Les détails des calculs utilisés pour ces déformations ainsi que des expérimentations relatives à ces techniques peuvent être trouvées dans Schaffer (1996) et Noik (1993).

## Zoom infini et zoom sémantique

Une autre façon de concilier détails et vue globale est de permettre à l'utilisateur de zoomer facilement : un zoom avant révèle les détails tandis qu'un zoom arrière révèle le contexte. Le zoom est une transformation géométrique à part entière et peut être plus ou moins complexe. Le zoom bitmap simple, qui consiste seulement

en un changement d'échelle est d'un intérêt limité quand il s'agit d'interagir avec un système de visualisation d'information car il n'est pas de nature à révéler suffisamment d'information. Ce qu'il faut au contraire, c'est un procédé qui permette une transformation géométrique (grossissement et recentrage) associée à une sémantique (chaque niveau de zoom correspond à un niveau différent de détails affichés) (Bederson et Hollan, 1994; Bartram et al., 1995). C'est l'approche choisie dans les interfaces zoomables avec le projet *Pad* (Perlin et Fox, 1993) et ses successeurs *Pad++* (Bederson et Hollan, 1994) et plus récemment *Jazz* (Bederson, 2000).

*Pad* offre une surface infinie sur laquelle sont présentées des informations et dans laquelle on peut zoomer infiniment. Chaque objet présent sur la surface a une position donnée et occupe un espace donné à un niveau de zoom donné. Pour naviguer sur cette surface l'utilisateur doit faire des zooms avant ou arrière et des déplacements XY appelés pans (pour panoramique). Les objets soumis à un zoom peuvent générer des présentations différentes en fonction du niveau de zoom courant. Par exemple, un calendrier peut être représenté par le mot « Calendrier » à un faible niveau de zoom ; à mesure que l'on zoome dessus apparaissent les années, les mois puis les jours. De nombreuses applications à ce type d'interaction sont envisageables. L'utilisation de *Pad* pour la visualisation d'historiques de navigation sur le WWW en est un exemple (Hightower et al., 1998). Ce travail a montré que la navigation multi-échelle dans un historique de navigation apportait des gains de performances significatifs en terme de temps passé pour accomplir une tâche donnée, ainsi qu'en terme de nombre de pages visitées en vain, pour certaines tâches de navigation.

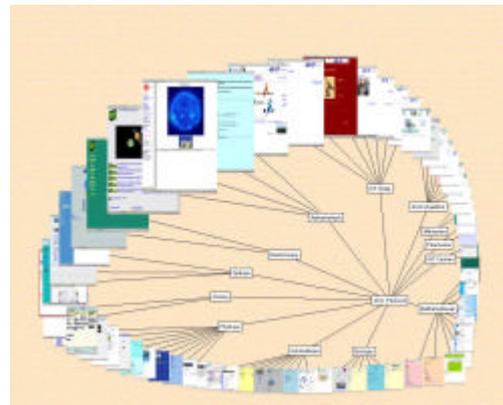
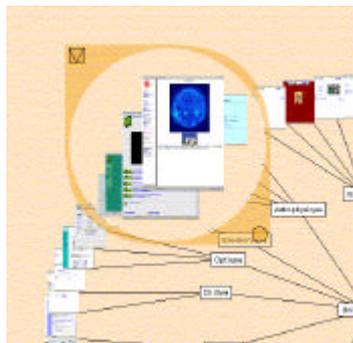
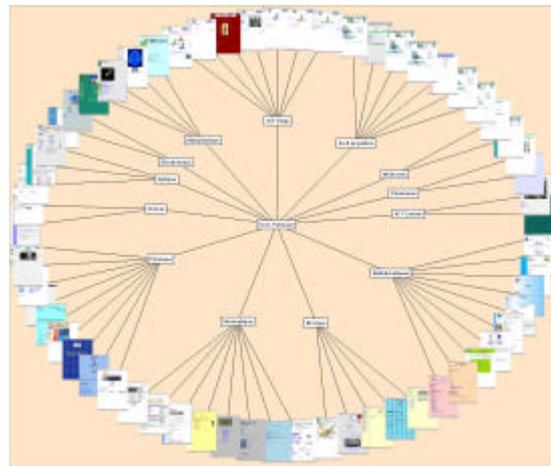


FIG. 17 – A partir de la vue du haut, fisheye global en coordonnées cartésiennes et fisheye local en coordonnées polaires en bas à gauche)

Cependant la navigation multi-échelle reste encore un type de navigation mal connue à plusieurs titres : temps des déplacements physiques, mise en place de stratégies de navigation typiques et facilement appréhendées par tous, orientation, etc. Les diagrammes espace-échelle de Furnas et Bederson (1995) permettent de mieux comprendre la navigation dans une interface zoomable. Ils affectent à l'échelle une dimension à part entière. Dans cet espace 3D (deux dimensions d'espace et une d'échelle), le plus court chemin entre deux points n'est plus la ligne droite, mais une trajectoire en V constituée d'un zoom arrière jusqu'à ce que la cible du déplacement soit visible suivi d'une succession de zooms avant et de pans afin de se rapprocher de la cible. Pourtant les utilisateurs n'ont pas de difficulté à mettre en œuvre ces trajectoires. En fait, un résultat remarquable (Guiard et al., 2001) est que le temps de pointage dans une interface zoomable suit exactement la

même loi de Fitts que pour le pointage normal sur écran. La navigation dans un espace zoomable est donc très efficace.

La quantité d'information que l'on peut mettre dans une interface zoomable est gigantesque. Aussi dans la pratique une très petite partie de l'espace est utilisée. En conséquence, il est facile de se perdre, ou au moins de se trouver dans une partie d'espace où il n'y a rien, le «desert fog» (Jul et Furnas, 1998). Aussi est-il nécessaire d'organiser l'espace de façon à limiter ces situations de désorientation.

## CONCLUSION

Si les travaux sur la recherche d'information sont relativement anciens, ils ont reçu un regain d'intérêt avec l'explosion du World-Wide Web et l'essor des moteurs de recherche. Cependant les moteurs de recherche actuels, aussi sophistiqués soient-ils, retournent habituellement une simple liste ordonnée de pages souvent trop nombreuses pour être toutes explorées. Ils doivent donc être complétés par des systèmes permettant l'exploration de l'espace documentaire. Ainsi, une recherche ciblée sera souvent suivie d'une phase exploratoire pour affiner la recherche, et une phase d'exploration permettra de mieux définir les termes d'une recherche ciblée.

Le domaine de la visualisation interactive d'information est plus récent que celui de la recherche d'information, même s'il est issu de domaines plus anciens comme la visualisation scientifique. L'avènement des postes de travail graphiques et l'augmentation de la puissance de calcul ont permis d'explorer et de développer des techniques interactives exploitant au mieux les capacités sensori-motrices et cognitives de l'être humain. Ainsi, la visualisation d'information cherche à présenter les données d'une façon telle qu'un utilisateur puisse en extraire un sens que la machine n'aurait pu découvrir. L'interaction en temps réel permet de faciliter la navigation dans l'espace d'information et ainsi de mieux comprendre la structure et les caractéristiques de cet espace. Cette meilleure connaissance permet ensuite de formuler des requêtes plus efficaces au système de recherche.

On peut identifier plusieurs directions de recherche dans ce domaine. Tout d'abord, de nouvelles techniques de visualisation apparaissent régulièrement pour des besoins ou des types de données particuliers. L'évaluation de ces techniques est également un problème encore largement ouvert. Cependant, c'est dans les techniques d'interaction que l'on peut attendre les développements les plus significatifs. En effet c'est au niveau de l'interaction que l'on peut améliorer le couplage entre techniques de visualisation et de recherche d'information. On a vu que de nombreuses techniques de visualisation et d'interaction avaient pour but de combiner vues globales (contexte) et vues locales (focus). Une vue locale peut être vue comme le résultat d'une requête sur une vue globale, auquel cas il y a tout intérêt à permettre le contrôle dynamique et interactif de la requête grâce à des techniques d'interaction avancées. Par exemple, l'interaction bimanuelle permet d'augmenter le nombre de degrés de libertés contrôlés par l'utilisateur simultanément. Cela peut servir à contrôler plusieurs dimensions simultanément dans une requête dynamique, à multiplexer des représentations différentes avec des lentilles magiques (Bier et al., 1994) ou à contrôler simultanément pan et zoom dans une interface zoomable.

Ainsi, recherche et visualisation interactive s'affirment comme des domaines complémentaires dont la combinaison offre plus que la somme des parties. Les travaux dans ce domaine, présentés dans cet article, sont amenés à se développer notamment pour mieux explorer cette complémentarité.

## REFERENCES

- Ahlberg, C. & Shneiderman, B. (1994). Visual Information Seeking : Tight Coupling of Dynamic Query Filters with Starfield Display. *Proc. ACM Human Factors in Computing Systems (CHI'94)*, pp. 313–318.
- Ahlberg, C., Williamson, C., Shneiderman, B. (1992). Dynamic queries for information exploration: An implementation and evaluation. *Proc. ACM Human Factors in Computing Systems (CHI'92)*, pp. 619–626
- Ahlberg, C. & Wistrand, E. (1995). IVEE: An information visualization & exploration environment. *Proc. IEEE Information Visualization'95*, pp. 66–73.
- Andrews, K. & Heidegger, H. (1998). Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs. *Proc. IEEE Symposium on Information Visualization (InfoVis'98)*, pp. 9–12.
- Baleyrier, L. (2001). *www.kartoo.com*.

Hascoët, M. and M. Beaudouin-Lafon (2001). "Visualisation interactive d'information." *Revue I3 I(1)*: 77-108.

- Bartram, L., Ho, A., Dill, J., Heningman, F. (1995). The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces. *Proc. ACM Symposium on User Interface Software and Technology (UIST'95)*, pp. 207–215.
- Battista, G. D., Eades, P., Tamassia, R., Tollis, I. G. (1999). *Graph Drawing: algorithms for the visualization of graphs*, Prentice Hall.
- Bederson, B., Meyer, J., Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *Proc. ACM Symposium on User Interface Software and Technology (UIST 2000)*, pp. 171–180.
- Bederson, B. H. & Hollan, J. D. (1994). Pad++: a zooming graphical interface for exploring alternate interface physics. *Proc. ACM Symposium on User Interface Software and Technology (UIST'94)*, pp. 17–26.
- Bertin, J. (1977). *La Graphique et le Traitement Graphique de l'information*, Flammarion.
- Bier, E. A., Stone, M. C., Fishkin, K., Buxton, W., Baudel, T. (1994). A Taxonomy of See-Through Tools. *Proc. ACM Human Factors in Computing Systems (CHI'94)*, pp. 358–364.
- Bruley, C. & Genoud, P. (1998). Contribution à une Taxonomie des Représentations Graphiques de l'Information, *Actes Dixièmes Journées Francophones sur l'Interaction Homme-Machine (IHM'98)*, AFIHM .
- Card, S. K. & Mackinlay, J. D. (1997). The Structure of the Information Visualization Design Space. *Proc. IEEE Symposium on Information Visualization (InfoVis'97)*, pp. 92–99.
- Card, S. K., Robertson, G. G., Mackinlay, J. D. (1991). The Information Visualizer, An Information Workspace. *Proc. ACM Human Factors in Computing Systems (SIGCHI'91)*, pp. 181–188.
- Chalmers, M. (1993). Using Landscape Metaphor to Represent a Corpus of Documents. *LNCS* , pp. 377–389.
- Chalmers, M. & Chitson, P. (June 1992). Bead : Explorations in Information Visualization. *Proc. ACM Conference on Information Retrieval (SIGIR'92)*, pp. 330–337.
- Chimera, R. (1992). Value bars: An information visualization and navigation tool for multiattribute listings. *Proc. ACM Human Factors in Computing Systems (CHI'92)*, pp. 293–294.
- Eick, S. G., Steffen, J. L., Sumner, E. E. (1992). SeeSoft- A tool for visualizing line-oriented software statistics. *IEEE Transactions on Software Engineering*, **18**(11):957-968.
- Furnas, G. & Bederson, B. (1995). Space scale diagrams: understanding multiscale interfaces. *Proc. Human Factors in Computing Systems (CHI'95)*, pp. 234–241.
- Furnas, G. W. (1986). Generalized Fisheye Views. *Proc. Human Factors in Computing Systems (CHI'86)*, pp. 16–23.
- Furnas, G. W. & Zacks, J. (1994). Multitrees: Enriching and Reusing Hierarchical Structure. *Proc. Human Factors in Computing Systems (CHI'94)*, pp. 330–336.
- Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*, Houghton-Mifflin (Boston).
- Guiard, Y., Bourgeois, F., Mottet, D., Beaudouin-Lafon, M., (2001). Beyond the 10-bit barrier: Fitts' law in multiscale electronic worlds. *Proc. IHM-HCI 2001*, Springer-Verlag, à paraître.
- Hascoët, M. (2001). Interaction and visualization supporting web browsing patterns. *Soumis à IEEE Symposium on Information Visualization (InfoVis 2001)*.
- Hascoët-Zizi, M. & Pediotakis, N. (1996). Visual Relevance Analysis. *Proc. ACM Conference on Digital Libraries (DL'96)*, pp. 54-62.
- Hearst, M. (1995). TileBars: Visualization of term distribution information in full-text information access. *Proc. ACM Conference on Human Factors in Computing Systems (CHI'95)*, pp. 59–66.
- Hemmje, M., Kunkel, C., Willett, A. (1994). LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval. *Proc. ACM Conference on Information Retrieval (SIGIR'94)*, pp. 249–259.
- Herman, I., Melançon, G. et al. (2000). Graph visualization and Navigation in Information Visualisation. *IEEE trans. on Visualization and Computer Graphics* **6**(1): pp. 24–43.
- Hightower, R., Ring, L. T., Helfman, J. I., Bederson, B., Hollan, J. (1998). Graphical multiscale web histories: a study of PadPrints. *Proc. ACM Conference on Hypertext and Hypermedia (HT'98)*, pp. 58–65.
- Jul, S. & Furnas, G. W. (1998). Critical Zones in Desert Fog: Aids to Multiscale Navigation. *Proc. ACM Symposium on User Interface Software and Technology (UIST'98)*, pp. 97–106.
- Kahn, P. (1999). Mapping Web Sites. <http://dynamicdiagrams.com/seminars/mapping/maptoc.htm>.
- Kamada, T. & Kawai, S. (1989). An Algorithm for Drawing General Indirect Graphs. *Information Processing Letters* **31**(1):7–15.
- Keim, D. A. & Kriegel, H. P. (1996). Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transactions on Knowledge and Data Engineering*, **8**(6):923–938.

Hascoët, M. and M. Beaudouin-Lafon (2001). "Visualisation interactive d'information." *Revue I3* 1(1): 77-108.

- Kleinberg, J. (1998). Authoritative sources in a Hyperlinked environment. *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pp. 668–677.
- Lamping, J., Rao, R., Pirolli, P. (1995). A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. *Proc. ACM Human Factors in Computing Systems (CHI'95)*, pp. 401–408.
- Leung, Y. K. & Apperley, M. D. (1994). A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction*, 1(1):126–160.
- Mackinlay, J. D., Robertson, G. G., Card, S. K. (1991). The Perspective Wall: Detail and Context Smoothly Integrated. *Proc. ACM Human Factors in Computing Systems (SIGCHI'91)*, pp. 173–179.
- Myers, D.G. (2000). *Psychology (6th edition)*, Worth Publishing.
- Noik, E. G. (1993a). Exploring Large Hyperdocuments : Fisheye Views of Nested Networks. *Proc. ACM Conference on Hypertext and Hypermedia (HT'93)*, pp. 192–206.
- Noik, E. G. (1993b). Layout-independent Fisheye Views of Nested Graphs. *Proc. IEEE Symposium on Visual Languages (VL'93)*, pp. 336–341.
- Olsen, K. A. et al. (1993). Visualization of a document collection: The VIBE System. *Information Processing & Management*, 29(1):69-81.
- Perlin, K. & Fox, D. (1993). Pad: An alternative An alternative Approach To The Computer Interface. *Proc. ACM SIGGRAPH'93*, pp. 57–64.
- Robertson, G. G. & Mackinlay, J. D. (1993). The Document Lens. *Proc. ACM Symposium on User Interface Software and Technology (UIST'93)*, pp. 101–108.
- Robertson, G. G., Mackinlay, J. D., Card, S. K.. (1991). Cone Trees: Animated 3{D} Visualizationz of Hierarchical Information. *Proc. ACM Human Factors in Computing Systems (SIGCHI'91)*, pp. 189–194.
- Salton, G. (1989). *Automatic Text Processing*, Addison-Wesley.
- Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*, Mc Graw Hill.
- Sarkar, M. & Brown, M. H. (May 1992). Graphical Fisheye Views of Graphs. *Proc. ACM Human Factors in Computing Systems (SIGCHI'92)*, pp. 83–91.
- Schaffer, D. (1996). Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Transactions on Computer-Human-Interaction* 3(2):162–188.
- Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Information Visualizations. *Proc. IEEE Symposium on Visual Languages (VL'96)*, pp. 336–343.
- Shneiderman, B. (1992). Tree Visualization with Tree-Maps: 2-D space-Filling Approach. *ACM Transactions on Graphics* 11(1):92–99.
- Smith, R. B. & Taivalsaari, A. (1999). Generalized and stationary scrolling. *Proc. ACM Symposium on User Interface Software and Technology (UIST'99)*, pp. 1–9.
- Tufte, E. R. (1983). *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Connecticut.
- Tweedie, L. (1997). Characterizing Interactive Externalizations. *Proc. ACM Conference on Human Factors in Computing Systems (CHI '97)*, pp. 375–382.
- Walker, J. (1990). A node positioning algorithm for general trees. *Software Practice and Experience* 20(7):685–705.
- Ware, C. & Lewis, M. (1995). The DragMag Image Magnifier. *Proc. ACM Human Factors in Computing Systems (CHI'95) Conference Companion*, pp. 407–408.
- Williamson, C. & Shneiderman, B. (1992). The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. *Proc. ACM Conference on Information Retrieval (SIGIR'92)*, pp. 338–346.
- Witten, I. H., Moffat, A., Bell, T. C. (1999). *Managing Gigabytes, second edition*, Morgan Kaufmann.
- Zizi, M. & Beaudouin-Lafon, M. (1995). Hypermedia Exploration with Interactive Dynamic Maps. *International Journal on Human Computer Studies*, 43:441–464.

Note : toutes les figures de cet article sont publiées avec la permission de leurs auteurs respectifs.