

# Visualisation interactive de documents

Mountaz Hascoët  
LIRMM, Univ. Montpellier II

Mountaz Hascoët, Univ. Montpellier II

1

## Contexte et motivations

- Essors de nouvelles techniques de visualisation
  - lié à l'augmentation des quantités d'information disponibles sous forme électronique
  - lié aux limitations des langages de requête traditionnels
    - exploration/navigation
- Pourquoi faire de la visualisation?
  - Aide à la recherche d'information
    - visualisation, exploration des résultats d'une requête
  - Aide à la découverte de nouvelles ressources
    - faire apparaître des relations, des structures, des chemins d'accès
  - Permettre la manipulation de cette information

Mountaz Hascoët, Univ. Montpellier II

2

1

# Plan

- I. Des techniques de visualisation par type de données à visualiser
- II. Evaluation
- III. Des techniques d'interaction spécifiques
- IV. Application

3

# Structures linéaires

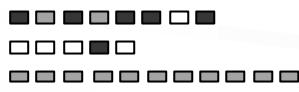
- Les données
  - données temporelles (une liste de documents par date) ou dans un ordre conventionnel (un annuaire de personnes)
- Techniques classiques
  - insuffisantes pour représenter des grandes listes
    - surface occupée,
    - lenteur du parcours,
    - proportion du visible/invisible.
  - limitées dans la représentation des index

4

## Structures linéaires

- Ajout d'indices visuels

– ex: Tilebars



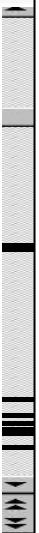
Un document globalement très pertinent  
Un document ponctuellement très pertinent  
Un document globalement moyennement pertinent

– ex: ValueBars

- Greeking

– ex: Seesoft

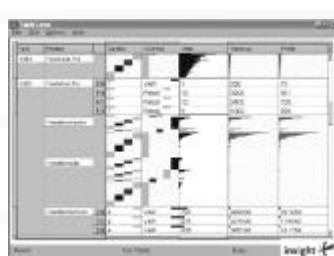
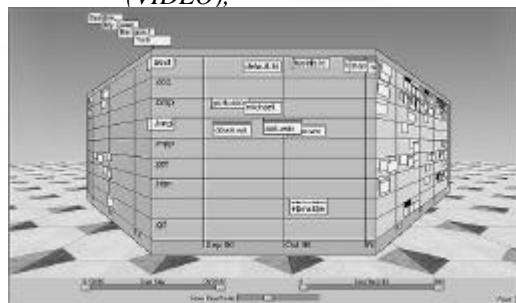
5



## Structures linéaires

- Les alternatives

– défilement à grain variable  
(Table Lens(Xerox))  
(VIDEO),



– 3D avec focus + contexte ( Le mur fuyant (Xerox))(VIDEO)

6

## Données spatiales

- Données pour lesquelles les coordonnées x,y, z ont un sens
  - sens concret ou abstrait
- Exemples en 2D
  - GIS,
  - ThemeMap, SemioMap, ...
- Exemples en 3D
  - Bureaux virtuels, WebBook, visualisation industrielle en VRML, etc



7

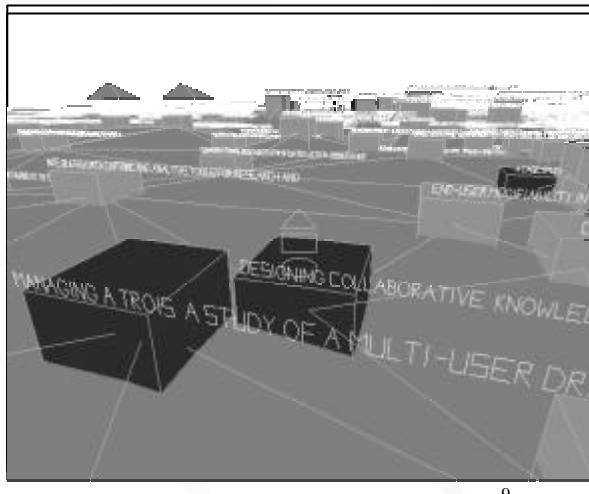
## Structures « vectorielles »

- Issues de la recherche d 'information
  - les informations (par ex des documents) sont représentées par des vecteurs
  - les similarités ou distances entre informations peuvent être définies comme des calculs sur ces vecteurs
- Techniques de visualisation
  - Métaphore du paysage (Chalmers 93)
  - Utilisation d 'un espace distance (Korfhage91)

8

## Structures vectorielles (suite)

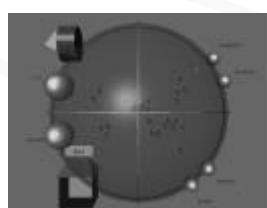
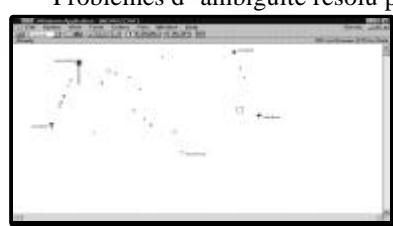
- Métaphore du paysage (Chalmers 93)
  - Les points d 'un espace 3D représentent les informations (par ex. des documents)
  - le placement de ces points dans l 'espace est significatif



9

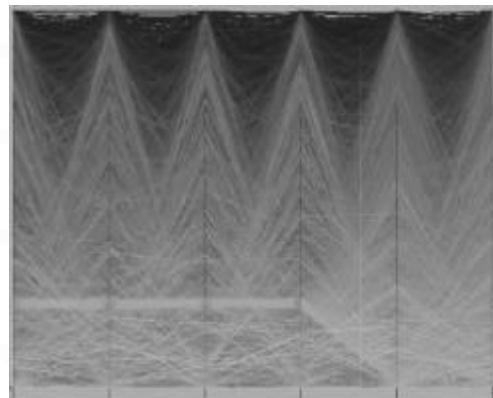
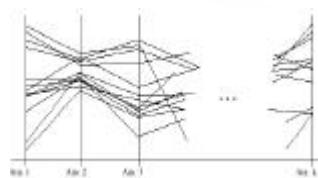
## Structures vectorielles (suite)

- Utilisation d 'un espace distance (Korfhage91)
  - Définition de points de référence
  - Placement des points représentant l 'information en fonction de leur distance par rapport aux points de référence
  - Originellement en 2D, extension à la 3D (GMD, Lyberwold)
  - Problèmes d 'ambiguïté résolu par animation et interaction



## Données multi-dimensionnelles

- Lignes parallèles  
(Inselberg)
  - k axes équidistants représentant chaque attribut
  - l 'échelle de chaque axe correspond aux valeurs min et max de chaque attribut
  - chaque item est représenté par une polyligne

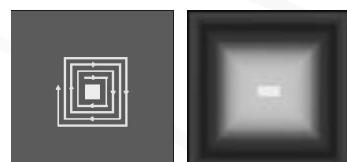
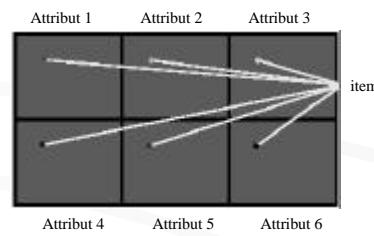


15 000 items avec une couleur qui dépend de la requête (extrait de Keim97)

11

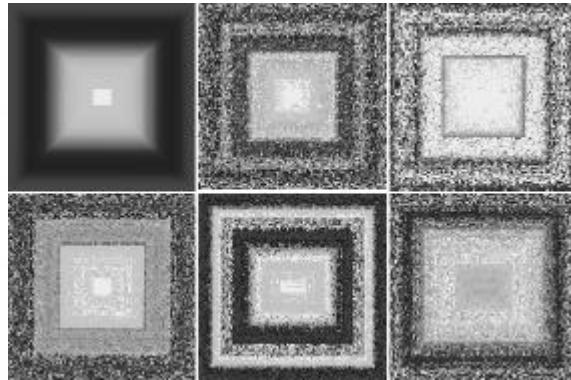
## Données multi-dimensionnelles

- Techniques orientées pixel [Keim91-98]
  - 1 pixel de couleur représente 1 attribut des données
    - chaque intervalle de valeur des attribut est associé à une couleur d'une table de couleur fixe
  - Chaque attribut est représenté dans une fenêtre séparée
  - Placement en spirale par ex.



12

## Techniques orientées pixel (suite)

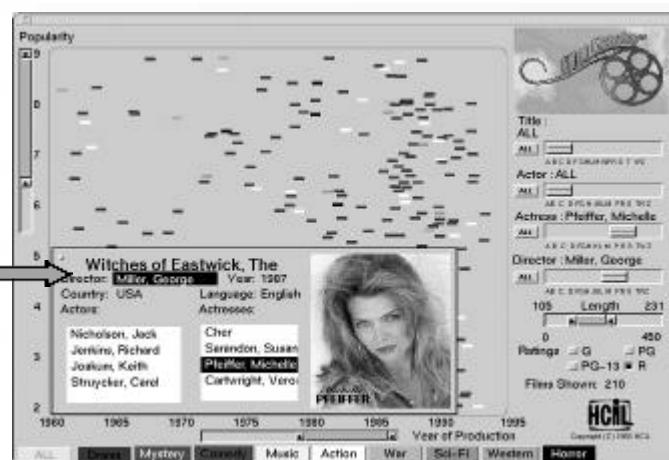


Résultats d'une requête complexe [Keim94]

13

## Données multi-dimensionnelles

- Starfield  
(Shneiderman,Ahlberg)
- axes personnalisables
- attributs graphiques (formes et couleurs) des items personnalisables



14

## Données multi-dimensionnelles

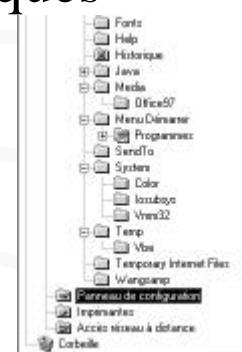
- Vues multiples
  - intégrer des vues multiples et totalement différentes de la même information
  - ex: Visage, Visible Human



15

## Structures hiérarchiques

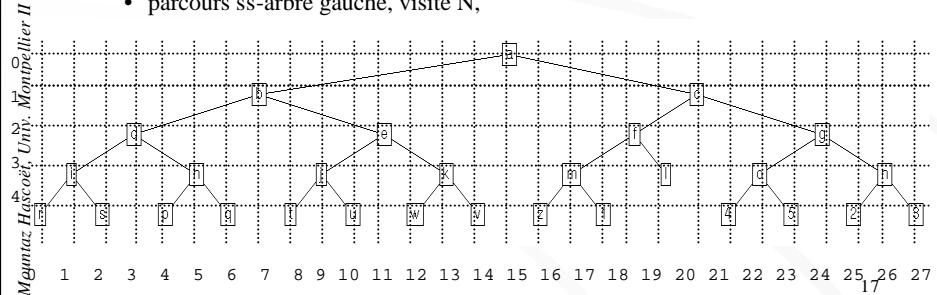
- Approche traditionnelle
  - limitée en terme de nombre de nœuds
  - accent mis sur les détails aux dépens du tout



16

## Visualisation d'arbres binaires

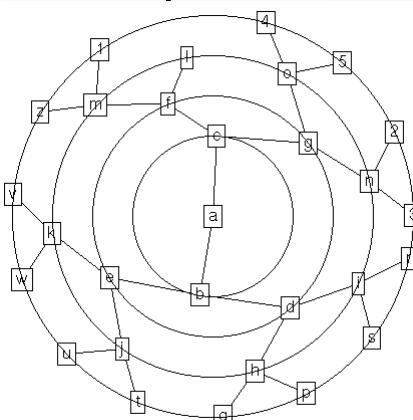
- Un algorithme simple
  - $y(N) = \text{profondeur}(n) * \text{espx}$
  - $x(N) = \text{rang}(N) * \text{espx}$ 
    - $\text{rang}(N)$  est le rang lors d'une traversée en profondeur ordre infixe
  - parcours ss-arbre gauche, visite N,
- Largeur non optimisée
  - Largeur = nbnoeud \* espx
- Parent centré entre fils gche et droit
- Dessin cohérent des sous-arbres isomorphes



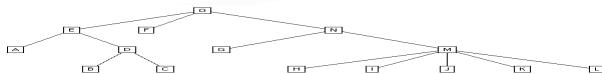
## Variations sur l'algorithme simple

- Adaptation de l'algorithme des arbres binaires
  - pour les arbres généraux
  - mêmes propriétés qu'avec l'algorithme de placement binaire

Adaptation pour dessin radial



18



## Dessin d'arbres optimisé

- Walker 1990 (extension de Reingold-Tilford 1983)
- Principes
  - Placement des sous-arbres comme des unités rigides
    - quand un nœud est déplacé tous ces descendants sont déplacés
    - l'arbre général est dessiné récursivement des feuilles vers la racine
  - Chaque nœud utilise une décomposition de son abscisse en 2 champs
    - abscisse préliminaire
    - modifieur
- Propriétés
  - Minimise la largeur de l'arbre
  - Dessin équivalent des sous-arbres isomorphes
  - Parent centré *au dessus* des ces enfants
- Remarque
  - calcul en coordonnées flottantes
  - minimiser la largeur du dessin est un problème NP-complet si on impose que les coordonnées soient entières

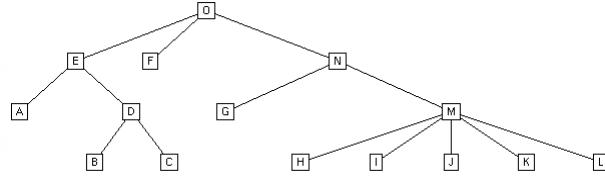
19

## Dessin d'arbres optimisé

- Walker (1990)
  - extension de Reingold-Tilford (83)
- Principes
  - Placement des sous-arbres comme des unités rigides
    - quand un nœud est déplacé tous ces descendants sont déplacés
    - l'arbre général est dessiné récursivement des feuilles vers la racine
  - Chaque nœud utilise une décomposition de son abscisse en 2 champs
- Propriétés
  - Minimise la largeur de l'arbre
  - Dessin équivalent des sous-arbres isomorphes
  - Parent centré *au dessus* des ces enfants
- Remarque
  - calcul en coordonnées flottantes
  - minimiser la largeur du dessin est un problème NP-complet si on impose que les coordonnées soient



## Algorithme pour un dessin optimisé (1/2)



### 1ère étape : parcours post-ordre

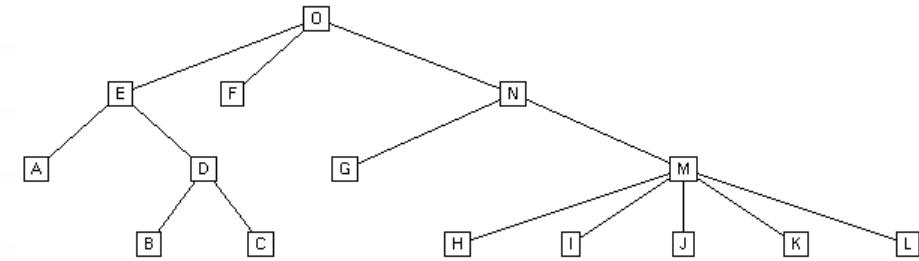
- calcul des coordonnées préliminaires et des modifieurs  
placement des ss-arbres de la racine indépendamment les uns des autres
- décalage des sous-arbres pour éviter recouvrements entre cousins



### 2ième étape: parcours pré-ordre

- calcul des coordonnées réelles:

21



- 1ère étape
  - **Parcours post-fixe de l'arbre**
  - à chaque visite d'un nœud on applique visite(N)

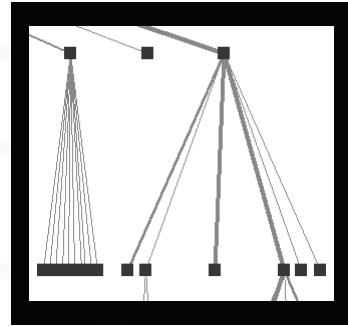
```

visite(N)
  si N a un frère gauche /*FG*/
    N.xprelim = FG.xprelim + tailleNoeud + espInterNoeuds;
    si N a des enfants /* D est le premier des enfants et G le dernier */
      N.modifier = N.xprelim - (D.xprelim + G.xprelim) / 2
    else /* N n'a pas de frère à gauche */
      si N a des enfants /* on le met directement au dessus des enfants */
        N.xprelim = (D.xprelim + G.xprelim) / 2
        N.modifier = 0
      else /* si N n'a pas d'enfants */
        N.xprelim = N.modifier = 0
  
```

22

## Ajout d'indices visuels lors de la visualisation de grands arbres

- Construction d'indices visuels
  - à partir de la structure de l'arbre
  - basé sur les nombres de Horton-Strahler
  - [Melançon99]
- Objectif
  - pour les arbres trop grands
    - pas forcément visibles entièrement
  - aide à l'orientation de l'utilisateur en donnant une forme graphique significative à l'arbre
    - par l'épaisseur des traits
    - la couleur (intensité)



23

## Ajout d'indices visuels

- Nombres de Horton-strahler
  - issus de l'étude des bassins hydrauliques
    - calcul d'une mesure de la forme que peuvent prendre des bassins hydrauliques organisés hiérarchiquement
  - définis pour des arbres binaires mais facilement généralisables aux arbres qcq

### • Définition pour les arbres binaires

- Pour un nœud  $N$  on note
  - $NHS(N)$  le nb de Horton-Strahler associé,
  - $G$  fils gauche de  $N$ ,  $D$  fils droit

$$\begin{array}{lll}
 NHS(N) & = 0 & \text{si } N \text{ n'a pas d'enfant} \\
 & = NHS(G) + 1 & \text{si } NHS(NG) = \\
 NHS(D) & = \max(NHS(NG), NHS(D)) & \text{sinon}
 \end{array}$$

24

## Visualisation d'arbre par cartes

- Principe simple
  - pour chaque nœud:
    - division verticale (horizontale) proportionnellement au nombre d'enfants et au poids de ces enfants
    - à chaque niveau on inverse l'orientation
- Algorithme
  - première passe: calcul des poids cumulés
  - deuxième passe: calcul des rectangles

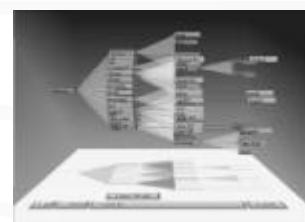


Un arbre vu par une carte,  
d'après shneiderman et al.92

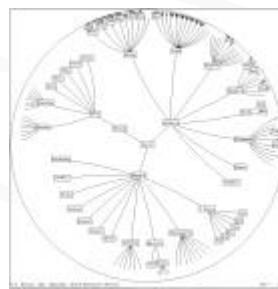
25

## Structures hiérarchiques (suite)

- Approche 3D
  - ✚ Plus de nœuds, plus de feuilles
  - ⬇ Problèmes liés à la 3D
    - occlusion, manipulation, orientation
    - (VIDEO)
- Géométrie hyperbolique
  - arbres hyperboliques (Xerox)
    - la surface du cercle croît exponentiellement avec son rayon
    - translation hyperbolique => navigation intuitive??
  - (VIDEO)



Arbre conique (Xerox)



Arbre hyperbolique (Xerox)

26

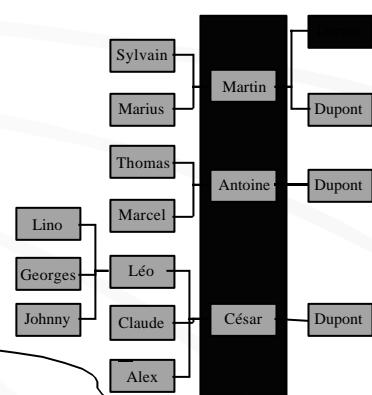
# Structures de multi-arbre

- Extension des structures hiérarchiques
  - Structure plus riche
- Définition multi-arbre
  - obtenu à partir d'un arbre en ajoutant un point d'entrée
- Propriétés
  - l'ensemble des descendants d'un nœud quelconque
    - appelé *arbre des contenus*
  - l'ensemble des ascendants forme un arbre
    - appelé *arbre des contextes*
  - Un multi-arbre n'est pas un arbre
    - Un nœud peut avoir plusieurs pères

# Structures de multi-arbre

- Des possibilités de visualisation intéressantes

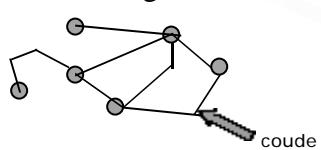
Présentation centrifuge et animation



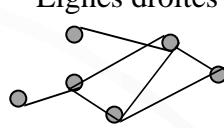
# Visualisation de graphes

- Algorithmes issus d'un domaine à part entière
  - dessin de graphes (graph drawing)
  - conventions de base
  - algorithmes
  - critères d'évaluation
- Conventions de base
  - 3 types de dessin admis pour les liens

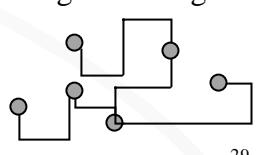
Multi-lignes



Lignes droites



Lignes orthogonales



29

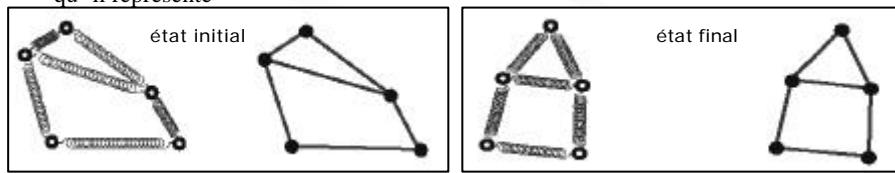
# Visualisation de graphes

- Nombreux cas
  - hypertextes, réseaux sémantiques, réseaux de ressources, etc.
- Familles d'algorithmes de visualisation de graphe
  1. Algorithmes s'inspirant d'un système de forces
    - Principe commun
      - définition d'un système de forces qui agissent sur les nœuds et les liens
      - trouver un état d'énergie minimum du système
        - » résolution de système d'équa. diff.
  2. Algorithmes par transformation du graphe
    - Principe commun
      - Rendre le graphe planaire (s'il ne l'est pas)
      - Utiliser un algorithme de placement de graphe planaire
    - Plusieurs versions
      - amélioration des performances
      - prise en compte du poids des nœuds et des liens

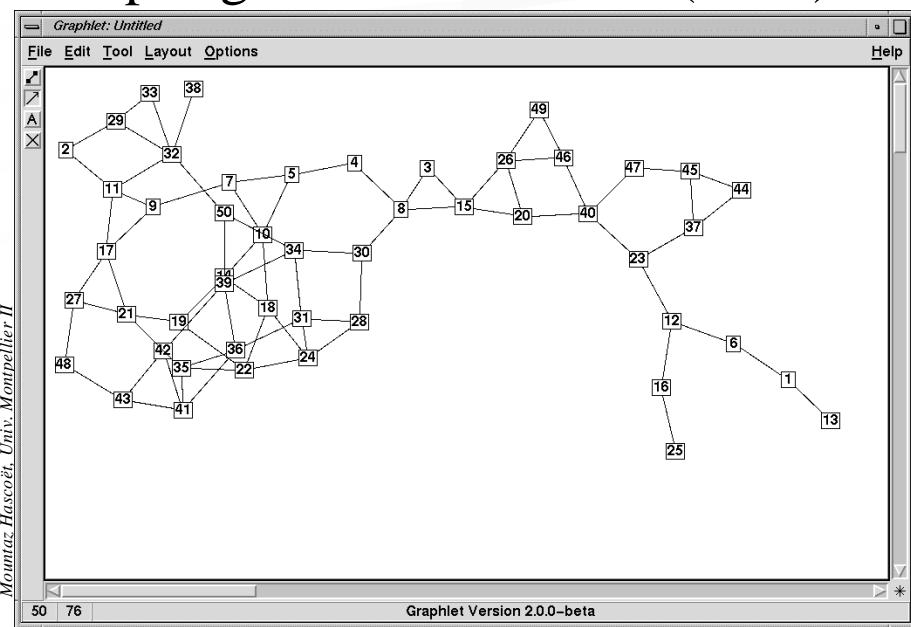
30

# 1. Algorithmes à base de forces (principes)

- Spring Embedder (Eades 1984)
  - un ressort de longueur unitaire représente chaque lien (ie couple de nœuds connectés)
  - un ressort de longueur infinie représente chaque couple de nœuds non connectés
  - un ressort comprimé repousse les nœuds qu'il représente
  - un ressort étiré rapproche les nœuds qu'il représente
- Spring 2 (Kamada&Kawai)
  - variante de Spring Embedder
  - la longueur associée à un ressort est proportionnelle à la distance entre les nœuds (plus court chemin entre les nœuds)
  - Le placement final tente de respecter ces distances
    - distances euclidiennes reflètent distances de graphe



## Spring - Kamada Kawai (1989)



# Point sur les algorithmes à base de systèmes de forces

- Avantages

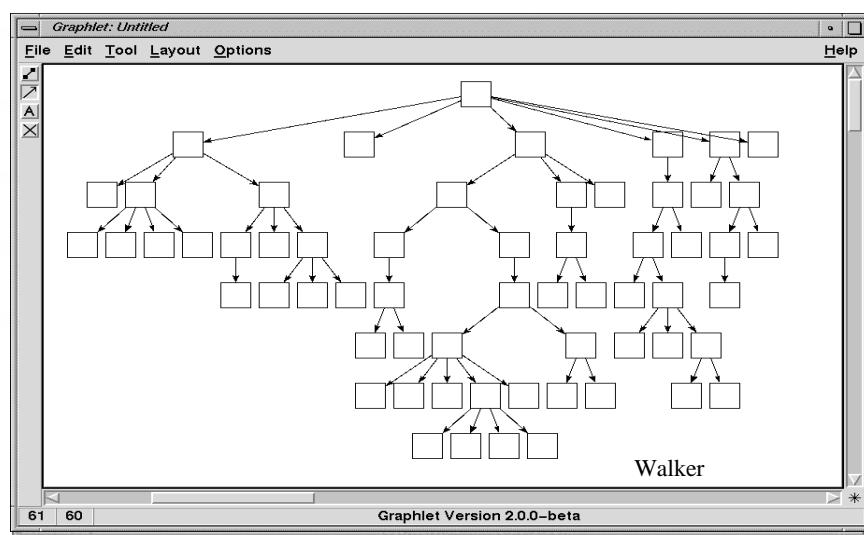
- Relativement simples à implémenter
- ajout d'heuristique facilement intégrable
- itératif => animations pour suivi
- conséquences positives de l'état d'équilibre
  - symétries
  - distances
- fonctionne bien pour de petits graphes

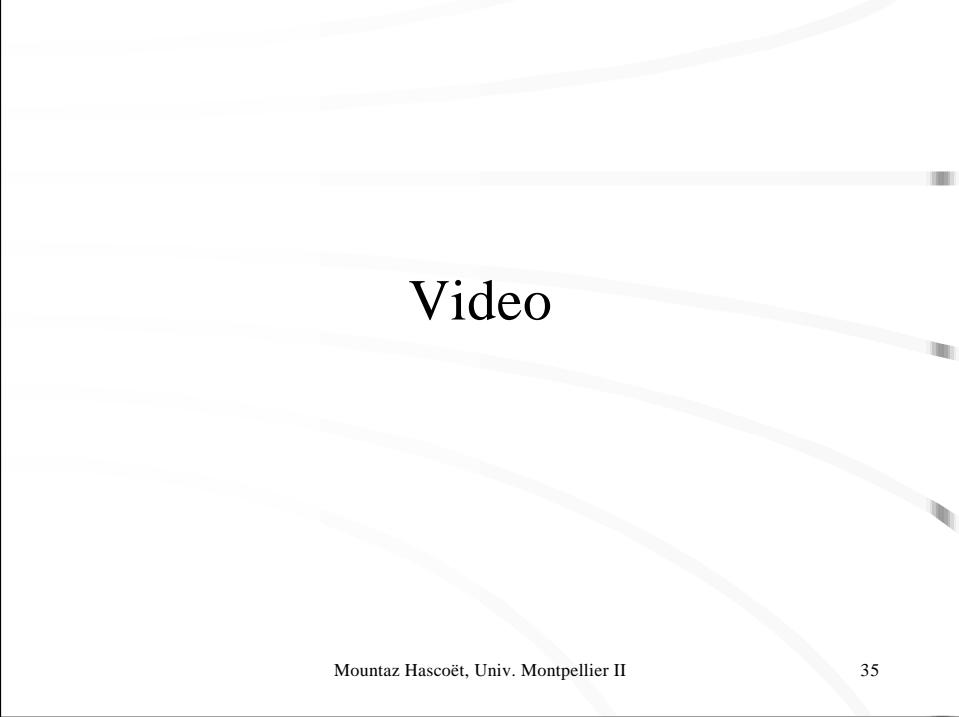
- Inconvénients

- problème de complexité
  - pour les graphes de taille élevée, calcul très long
- peu de résultats théoriques sur la qualité du dessin produite
- peu de contraintes prises en compte
  - exemple contraindre à une taille donnée

33

## Exemples Placements différents d'un même arbre

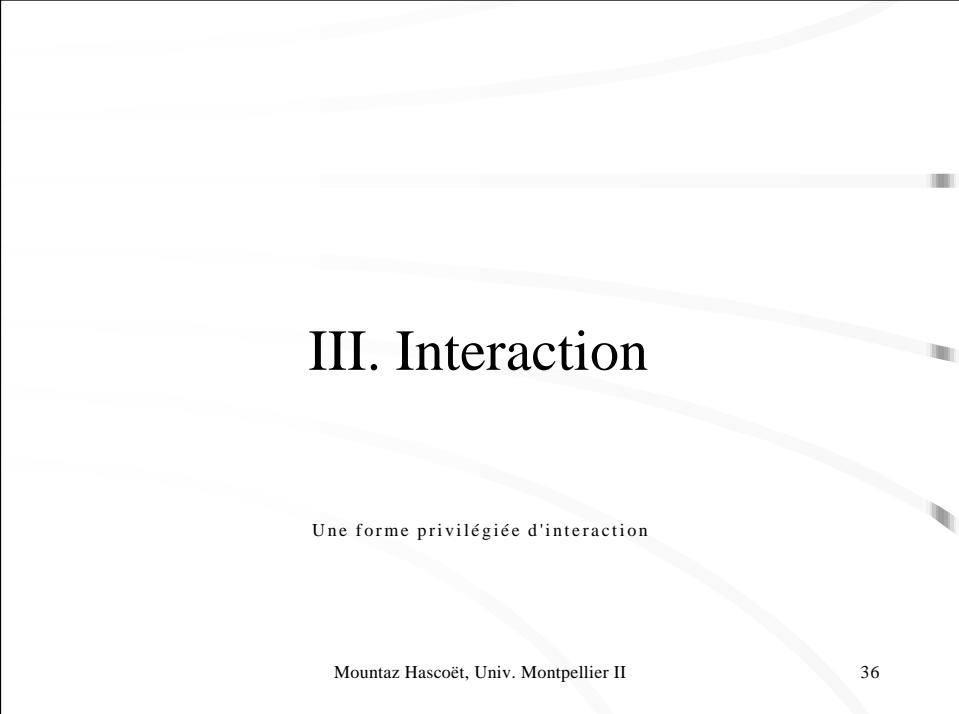




## Video

Mountaz Hascoët, Univ. Montpellier II

35



## III. Interaction

Une forme privilégiée d'interaction

Mountaz Hascoët, Univ. Montpellier II

36

# "Overview first, zoom and filter and details on demand"

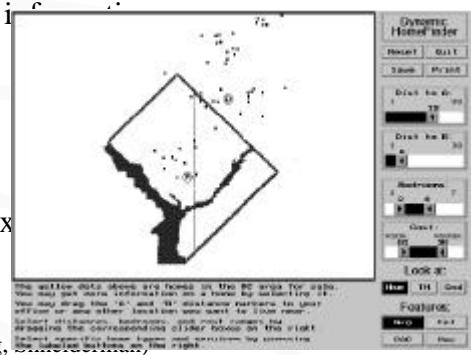
B. Shneiderman

- Filtrage dynamique
- Déformer interactivement
  - Zoom
    - Zoom sémantique
      - changement d 'échelle => changement de niveau de détails
    - Zoom infini (pad++)
  - Techniques de déformation
    - montrer des détails en gardant le contexte d 'exploration présent

37

## Filtrage dynamique

- Filtrage (type BD)
  - filtrage en temps réel des informations présentes à l'écran
  - fort couplage entre les données et leur présentation
  - temps de réponse cruciaux
  - Application
    - Requêtes dynamiques
      - Spotfire, IVEE (Ahlberg, ...)
    - Prévisualisation de requêtes



38

# Zoom

- cf cours zoom

# Distortion

- cf cours distortion