

# Navigation dans une base d'objets 3D

R. Trichet

J. Ricard

B. Tellez

A. Baskurt

LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information)

UMR 5205 CNRS/Université Claude Bernard Lyon 1

43, Boulevard du 11 novembre 1918, 69622 Villeurbanne Cedex - France

{rtrichet, jricard, btellez, abaskurt}@liris.cnrs.fr

## Résumé

*Nous abordons dans cet article le thème de la navigation dans une base d'objets 3D. Le système proposé est basé sur les descripteurs ART 3D. Il utilise un algorithme original de partitionnement par division qui donne des résultats sensiblement meilleurs que les nuées dynamiques floues pour un temps d'exécution nettement plus rapide. Pour la navigation, la base est structurée en un arbre hiérarchique en utilisant itérativement cet algorithme. Nous proposons également un système de visualisation 3D inspiré d'arbres de cônes permettant, à tout instant à l'utilisateur de visualiser le parcours de la base. La recherche s'effectue par similarité entre la forme de l'objet recherché et celles des objets de la base. Des tests ont été réalisés et montrent les avantages de l'approche proposée.*

## Mots clés

Navigation, Objets 3D, Classification non-supervisée, Visualisation 3D

## 1 Introduction

La nature des documents numériques a profondément évolué au cours des dernières décennies. Quelque en soit le type, (texte, image, son,...), leurs moyens de création, de duplication et de transmission se sont rapidement développés, conduisant leur nombre à s'accroître considérablement. En conséquence, la gestion, la recherche et l'exploration de ces documents nécessitent des moyens toujours plus performants. Cette constatation a conduit à décrire les documents en fonction de leur contenu multimédia notamment visuel. Dans ce contexte, la navigation a pour but de répondre au problème de la recherche d'un document précis ou d'un type de document en catégorisant et structurant la base à laquelle il appartient.

Ces travaux s'inscrivent dans le cadre du projet SEMANTIC-3D et s'effectuent au travers de la conception d'un prototype d'application industrielle traitant des objets 3D représentant des pièces mécaniques. A cause du volume des données à traiter d'une part, et des difficultés d'indexation d'autre part, la navigation sur des bases d'objets 3D n'est que rarement abordée dans la littérature. Tou-

tefois, il est possible de s'inspirer des solutions employées pour les bases d'images dont le fonctionnement est analogue. En effet, que ce soient des images ou des objets 3D, ils peuvent être décrit par des vecteurs descripteurs et traités de manière similaire. Les méthodes de recherche d'images par le contenu procèdent, dans un premier temps, à une indexation de la collection, autrement dit, à l'extraction des descripteurs bas niveaux des images (couleur, texture, forme,...) afin de pouvoir établir un critère de similarité entre elles. Par la suite, pour chaque requête de l'utilisateur, une distance est calculée en fonction de ces descripteurs et les images les plus proches de la requête sont retournées. Les deux problèmes majeurs de ce type d'exploration d'une base d'images à priori inconnues sont : la difficulté à appréhender la base dans sa globalité et l'absence de représentation des relations entre les images. Les méthodes de classification automatique apportent une aide précieuse dans la résolution de ce type de problème en découpant la base en groupes d'images similaires. Une fois les images de la base regroupées, plutôt que de représenter la totalité de la base, les systèmes de visualisation choisissent de définir une image résumée pour chaque groupe. Ainsi, l'utilisateur peut naviguer rapidement et efficacement dans la base d'image grâce à ces images résumées ne conservant que l'essentiel de l'information. Outre la structuration de la base, le choix de l'espace de visualisation est également important. Nous montrerons que la navigation dans une collection d'objets à l'aide d'une interface en trois dimensions peut également apporter à l'utilisateur une aide significative dans la visualisation de la collection dans son ensemble. En effet, en plus de la représentation d'objets 3D, l'ajout d'une dimension supplémentaire permet d'afficher un plus grand nombre d'informations.

Dans cet article, nous proposons une solution complète pour la navigation dans une base d'objets 3D. Après un état de l'art sur la navigation, nous présenterons notre approche qui propose un nouvel algorithme de classification par division, ainsi qu'un système de navigation 3D. Ce dernier est basé sur une interface utilisant des arbres de cône. Enfin, nous concluons en présentant les résultats obtenus, le prototype et les perspectives de notre travail.

## 2 Etat de l'art

Comme nous l'avons indiqué, un processus de navigation peut se décomposer selon trois phases :

- l'indexation qui permet de représenter les objets par un ensemble de descripteurs qui facilitent leur comparaison. Ces descripteurs permettent de représenter l'objet comme un point dans l'espace de représentation.
- La classification qui structure l'espace en découpant le nuage de points créé en groupes cohérents.
- La visualisation qui permet de consulter la disposition globale ou relative des objets et les informations qui leur sont associées.

## 3 Indexation

Avant toute analyse, il est nécessaire de procéder à l'indexation de la base, autrement dit à l'extraction de descripteurs bas niveaux de chaque objet. Le choix des descripteurs et leur nombre constituent une décision cruciale pour les futurs traitements à effectuer sur la base. Les descripteurs doivent être pertinents pour le type d'objet considéré dans la collection. Par exemple, le choix d'un descripteur de couleur ne sera d'aucune aide pour un ensemble d'images de couleurs relativement uniformes. De plus, ils doivent éviter la redondance d'information (non corrélation des descripteurs), inutile et qui fausse leur importance relative. Enfin, l'utilisation d'un trop grand nombre de descripteurs fournira certes une information plus précise mais plus fragmentaire et difficile à interpréter. Dans une collection, on peut distinguer deux types de données : les données homogènes et les données hétérogènes suivant que celles-ci soient de même type ou non (texte, nombre,...). Nous ne traiterons ici que de données homogènes. Il est à noter que les images 2D ou des objets 3D sont souvent décrits par des nombres. Dans ces cas, l'algorithme de classification est applicable quelque soit le type d'objet traité.

### 3.1 Classification

La classification a pour but de découper automatiquement le nuage de points créé en groupes de telle sorte que les éléments d'un même groupe soient les plus similaires possibles et que les éléments de groupes différents soient les plus dissemblables possibles. On distingue deux catégories de méthodes : la classification hiérarchique et le partitionnement. Les algorithmes de classification hiérarchique divisent le nuage de points en groupes pouvant être représentés par un arbre hiérarchique ou un dendrogramme (les feuilles représentant les points). Les méthodes de classification hiérarchique ont l'avantage de ne pas souffrir de problèmes d'initialisation et de créer des partitions aisément structurables mais présentent de nombreux inconvénients :

- elles ne tiennent pas compte du nuage de points dans sa globalité mais seulement des proximités immédiates de chaque point ;

- ce sont des méthodes statiques, autrement dit une fois un point attribué à un groupe, il est impossible de le ré-affecter à un autre groupe.

Plusieurs méthodes de ce type sont recensées : Craver [1] proposa en 1999 d'utiliser des arbres de division, Chen et Bouman présentèrent en 2000 [2], une méthode fortement inspiré du quad-tree et Kohonen [3] développa en 1995 les cartes auto-organisées ou cartes de Kohonen dans le cadre de nombreux systèmes de navigation. Les algorithmes de partitionnement, par opposition aux algorithmes de classification hiérarchique divisent un nuage de points en groupes sans aucune relation entre eux. L'algorithme le plus connu, les nuées dynamiques, assume l'hypothèse que le nombre de groupes est connu et affecté à chacun un centre de classe. Des lors, la partition retenue est celle qui minimise la distance des points de chaque classe à leur centre. Tout d'abord, l'algorithme est initialisé en plaçant aléatoirement les centres de classes. Puis, tant que le système n'est pas stabilisé, chaque point est ré-alloué à la classe dont le centre est le plus proche et le centre de chaque groupe est placé à son centre de gravité. Cet algorithme nécessite de connaître le nombre de groupes et est un algorithme rigide : chaque point ne peut être affecté qu'à une classe et une seule. Ce dernier défaut conduisit Bezdek [4] en 1981, à développer le partitionnement flou. Dans ce type d'algorithme, à tout point est attribué un pourcentage d'appartenance à chaque groupe. Si celui-ci dépasse un certain seuil, préalablement fixé par l'utilisateur, le point est attribué à ce groupe. Par la suite, les algorithmes de partitionnement évoluèrent pour pouvoir fonctionner avec un nombre de groupes à priori inconnus. On en distingue 2 catégories : le partitionnement par agglomération comme l'agglomération compétitive [5] ou l'ARC (adaptative robust compétition) proposé par Le-Saux en 2004 [6], qui considère les points comme autant de groupes isolés. Il élimine au fur et à mesure des itérations les groupes non pertinents en minimisant une fonction objectif jusqu'à l'obtention du nombre correct de groupes. Le partitionnement par division au contraire ajoute un nouveau groupe à la partition et réajuste celle-ci tant qu'elle ne satisfait pas à un critère d'optimalité. Le Principal Direction Divisive Partitionning (PDDP) proposé par Boley en 1998 [7] utilise les propriétés géométriques de l'analyse en composantes principales pour diviser itérativement le nuage de points. Les nuées dynamiques bissectives par contre est une variante des nuées dynamiques. Débutant avec deux groupes, il augmente itérativement le nombre de groupes et relance l'algorithme des nuées dynamique tant qu'un critère d'optimalité n'est pas satisfait.

### 3.2 Système de visualisation

Une fois la base catégorisée et structurée, il est nécessaire de représenter les informations obtenues de façon à les rendre facilement accessibles par l'utilisateur. L'interface est donc la finalisation de la conception d'un système de navigation. Elle doit permettre à l'utilisateur de visuali-

ser le maximum d'informations apportées par les étapes précédentes, d'intuitivement comprendre les mécanismes mis en oeuvre par le système et être ergonomique.

Les collections d'objets étant trop grandes pour être affichées dans leur intégralité, il faut choisir un objet-résumé pour représenter chaque groupe. De plus il faut représenter en faible dimension (en pratique 1,2 ou 3 dimensions) les données tout en limitant la perte d'information.

Le système de visualisation le plus largement répandu est la carte 2D (figure 1), où les images sont représentées par des vignettes réparties sur un plan, la distance entre celles-ci représentant la dissemblance entre les images. Les groupes étant structurés selon un arbre hiérarchique, chaque vignette est l'image représentative d'un sous-groupe, cliquer sur une vignette permet d'accéder au sous-groupe correspondant. Moghaddam [8] montra que ce type de visualisation apporte une meilleure compréhension des relations entre les images. Par contre, l'absence de représentation de la structure de navigation empêche l'utilisateur d'appréhender celle-ci dans sa globalité. En effet, si celui-ci effectue un mauvais choix au cours du processus de navigation, dans la majorité des cas, il recommencera le processus du début plutôt que de revenir en arrière. Les travaux récents de Santini en 2001 [9] ou de Moghaddam en 2004 [8] visent à permettre l'interactivité entre l'utilisateur et le système.



Figure 1 – Exemple de carte 2D affichant des images [10].

L'ajout d'une troisième dimension comporte de nombreux avantages : ceci rend l'affichage d'objets 3D possible, permet l'affichage d'un plus grand nombre d'objets sans surcharger l'écran et « détruit » moins l'information. Toutefois, une compréhension difficile des relations entre les objets et une interface souvent moins facile à prendre en main rebutent généralement les utilisateurs. Les interfaces 3D proposées jusqu'à présent sont les arbres de cônes développés par [11], FSN de [12] dont l'organisation hiérarchique est directement inspirée de celle de villes (bâtiments reliés par des routes) et Pathfinders Networks de [13] inspirés de représentations 3D utilisées en psychologie. La figure 2 montre des exemples de ces trois approches.

## 4 Notre approche

Pour répondre à la problématique de la navigation dans une base d'objets 3D, nous proposons un système complet de navigation. En effet, il suit le schéma classique indexation/classification/visualisation présenté dans la partie 2 et répond à chacune de ces nécessités.

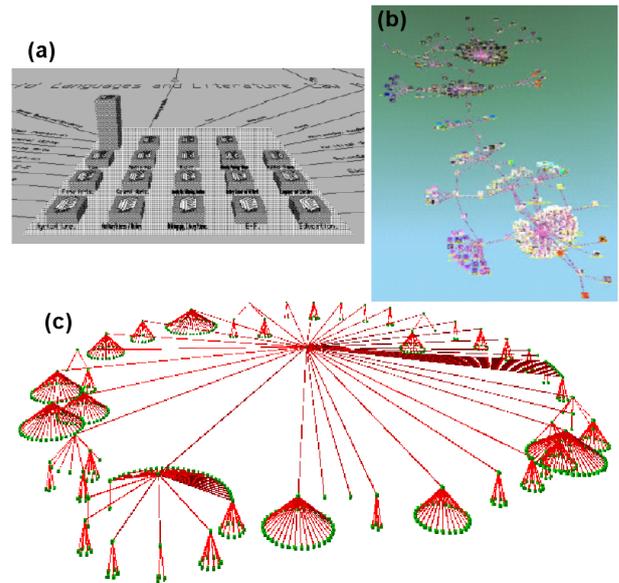


Figure 2 – Exemple d'affichage d'une base d'objets en trois dimensions. (a) FSN [12], (b) Pathfinders Networks [13] et les arbres de cônes [11].

### 4.1 Indexation

Le jeu de descripteurs utilisé ici sont les 74 descripteurs de forme ART 3D développés par Ricard [14] dans le cadre du projet SEMANTIC 3D et inspirés des descripteurs ART 2D utilisés par la norme MPEG-7. C'est un descripteur de forme 3D par approche région basé sur les moments, de faible taille et robuste aux principales transformations subit par les objets 3D.

### 4.2 Classification

Dans le cadre du projet SEMANTIC 3D, notre système de classification doit obéir à certaines contraintes : le nombre de groupes effectif est inconnu et la forme des groupes ainsi que celle du nuage de points initial sont quelconques. Typiquement, l'algorithme des nuées dynamiques bissectives répond à ces contraintes.

Il présente toutefois certains désavantages, notamment le fait que la qualité des groupes créés et la rapidité de convergence de l'algorithme dépendent du placement initial des centres de groupes. Nous avons choisi d'adapter cet algorithme en utilisant les propriétés géométriques de l'analyse en composantes principales pour guider le placement des centres de groupes. En effet, la première composante principale d'un nuage de points suit l'allongement maximal du nuage et passe par son centre de gravité (figure 3). Cette heuristique fonctionne dans la grande majorité des cas et permet de gagner un temps de calcul important. Pour déterminer la première composante principale, nous utilisons l'heuristique de Faloutsos et Lin [15] permettant d'approcher cette composante en temps linéaire, selon l'algorithme :

- choix aléatoire d'un point  $A$  dans le nuage de points ;

- détermination de  $O_1$  tel que la distance entre  $A$  et  $O_1$  soit maximale ;
- détermination de  $O_2$  tel que la distance entre  $O_1$  et  $O_2$  soit maximale 4 ;
- retourner  $(O_1, O_2)$ .

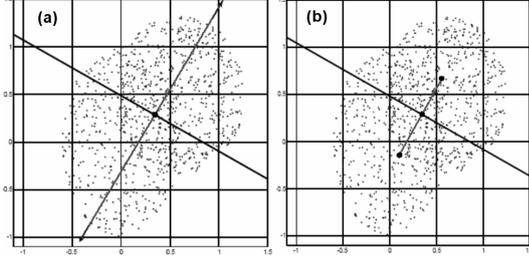


Figure 3 – La première composante principale permet de diviser le nuage de points en 2 groupes (a) et de définir leurs centres de gravité comme centre des groupes (b).

Le critère d'arrêt usuellement défini consiste à évaluer la qualité de la partition. Si celle-ci est inférieure à un seuil donné, les groupes sont considérés satisfaisants et l'algorithme s'arrête. Notre apport consiste à définir un critère pour chaque groupe, ce qui comporte plusieurs avantages :

- la qualité du partitionnement est évaluée de façon plus précise puisque chaque groupe est évalué indépendamment ;
- lorsqu'un groupe a une qualité satisfaisante, il n'est plus nécessaire de le traiter à nouveau ce qui permet de diminuer le temps de calcul pour des résultats identiques ;
- on peut ainsi obtenir des groupes de tailles différentes.

Nous avons choisi de représenter la qualité de notre partitionnement par l'énergie intra-groupe  $E_{intra}$ , qui caractérise la compacité du groupe et par l'énergie inter-groupe  $E_{inter}$ , qui décrit l'isolement du groupe par rapport aux autres groupes. La qualité d'un groupe se définit comme la somme pondérée des deux énergies selon la formule :

$$S(c) = \lambda_1 \cdot E_{intra} + \lambda_2 E_{inter} \quad (1)$$

où  $\lambda_1$  et  $\lambda_2$  sont les poids accordés respectivement à l'énergie intra et inter classes. Ils représentent l'importance relative de chacune des deux énergies.

**L'énergie intra-groupe** d'un groupe  $C$  est définie par :

$$E_{intra} = f(N_C) \sqrt{\frac{\sum_{i=0}^{N_C} \|x_i - \beta_c\|^2}{\frac{N_C}{2} \max_i \|x_i - \beta_c\|^2}} \quad (2)$$

où  $N_C$  est la cardinalité du groupe, et  $\beta_C$  le centre du groupe. Ce critère est évalué en comparant la distance moyenne des points au centre à celle, divisée par 2, du point le plus éloigné du centre de classe. Il sera élevé si la majorité des points est réparti entre le centre et cet axe médian, autrement dit pour les groupes compacts. Mais ce critère n'est plus représentatif pour les groupes à faible cardinalité, c'est pourquoi il est pondéré par la fonction  $f$  qui limite son importance lorsque la cardinalité du groupe est

faible. Pour notre système de navigation, nous avons défini empiriquement la fonction  $f$  pour qu'elle respecte le profil de courbe attendu.

**L'énergie inter-groupe** d'un groupe  $C$  est défini par :

$$E_{inter} = \frac{1}{N} \sum_{i=0}^N \sqrt{\frac{\frac{\sum_{j=0}^{N_C} \|x_j - \beta_C\|^2}{N_C} + \frac{\sum_{j=0}^{N_A} \|x_j - \beta_i\|^2}{N_i}}{\|\beta_C - \beta_i\|^2}} \quad (3)$$

où  $N$  est le nombre de groupes avec lesquels le groupe  $C$  est comparé,  $\beta_C$  et  $\beta_i$  les centres de classe respectifs des groupes  $C$  et  $i$ ,  $N_C$  et  $N_i$  leurs cardinalités respectives.

La distance entre deux groupes est évaluée en comparant l'écart entre les distances moyennes des points de ces deux groupes à leur centre et l'écart entre les centres de classe de ces deux groupes (Figure 4). Ainsi, deux groupes volumineux juxtaposés seront considérés comme éloignés si les distances entre les centres de classe seules étaient considérées, alors que l'utilisation de notre critère de proximité permet de considérer ces deux groupes comme plus ou moins proches en fonction de leur compacité.

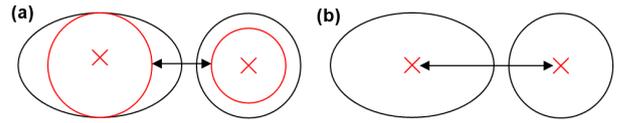


Figure 4 – Les croix sont les centres de classe, les ellipses les groupes et les cercles la distance moyenne des points aux centres des groupes. (a) distance basée sur la moyenne des points à leur centre respectif. (b) distance basée sur les centres de groupe.

Le critère inter-groupes est calculé en effectuant la moyenne de la distance entre le groupe considéré et les  $N$  groupes qui lui sont les plus proches. Dans notre système de navigation, nous avons estimé que cinq groupes voisins suffisent pour déterminer efficacement ce critère. La procédure complétée est détaillée dans l'algorithme 1.

Le principal avantage de cet algorithme est sa rapidité d'exécution et son principal inconvénient l'absence de partitionnement flou. Ce désavantage peut être pallié par une exécution ultérieure des nuées dynamiques floues à moindre coût car les centres de classes sont déjà positionnés. Cet algorithme est appliqué itérativement sur les centres de groupes issus de la précédente classification jusqu'à ce que le nombre de groupes soit inférieur à un seuil. La base est alors organisée en une structure d'arbre hiérarchique.

## 5 Navigation

Notre interface est une interface 3D inspirée des arbres de cônes [11]. Chaque groupe est modélisé par un cône, son représentant étant le sommet, et les objets ou les représentants des sous-groupes le composant étant modélisés sur le pourtour du cône. La similarité entre

---

**Algorithme 1** Algorithme de classification

---

**ENTRÉES :**  $E_N$  ensemble de  $N$  points,  $G$  ensemble de groupes,  $\alpha$  le seuil de qualité d'un groupe et  $Critere(G_i)$  la fonction dévaluation du critère d'arrêt sur le groupe  $G_i$

**SORTIES :**  $E$  ensemble de  $k$  partitions de  $E_N$

$E = \{\}$   $G = E_N$  et  $X = G_0$

**tantque**  $G \neq \{\}$  **faire**

Détermination de la 1ere composante principale de  $X$

Répartition des points dans les 2 groupes  $X_1$  et  $X_2$

Calcul des centres de gravité de  $X_1$  et de  $X_2$

$G \leftarrow X_1$  et  $G \leftarrow X_2$

Nuées dynamiques bissectives sur les groupes de  $G$

$C_{max} = 0$

**pour tout**  $G_i$  tel que  $i = 1..N$  **faire**

**si**  $Critere(G_i) < \alpha$  **alors**

$G = G - G_i$  et  $E = E + G_i$

**sinon si**  $Critere(G_i) > C_{max}$  **alors**

$X = G_i$  et  $Critere_{max} = Critere(G_i)$

**fin si**

**fin pour**

**fin tantque**

---

les objets est représentée par la proximité entre les objets sur la surface de ce cône. Afin d'éviter la superposition d'objets très similaires, voir identiques, un écart minimum entre les objets est imposé. Enfin, pour plus de clarté, aucun objet n'est modélisé dans le tiers supérieur du cône : leur sélection conduit à représenter les cônes très proches les uns des autres et nuit à leur visualisation. Visuellement, des couleurs sont utilisées pour distinguer les différentes entités de l'interface : en gris les représentants de groupes, en vert les feuilles et en rouge l'objet actuellement sélectionné.

Le fonctionnement de l'interface est le suivant : à tout instant sont représenté : l'objet sélectionné, ses objets fils (i.e les objets contenus dans le groupe dont il est le représentant), les objets constituant le chemin depuis le sommet de l'arborescence jusqu'à lui, et ses objets frères (i.e les autres objets constituant le groupe dont son père est le représentant). L'interface est initialisée par la modélisation de l'objet au sommet de l'arborescence ainsi que de ses fils. Par la suite l'utilisateur navigue dans la base en sélectionnant, à chaque étape l'objet affiché qui lui semble le plus proche de celui qu'il cherche. Il dispose pour cela de 2 moyens : cliquer avec la souris sur l'objet d'intérêt ou utiliser les flèches du pavé numérique. Lorsqu'un objet est sélectionné (figure 5) la fenêtre est centrée sur celui-ci, une rotation est effectuée autour de son cône afin de le placer à l'avant plan et les objets adéquats sont ajoutés ou ôtés afin de respecter la règle d'affichage énoncée plus haut.

Ce type d'interface permet à la fois de pouvoir afficher des objets en 3D et de visualiser le chemin suivi dans l'arborescence tout en restant claire et facile d'utilisation. Son

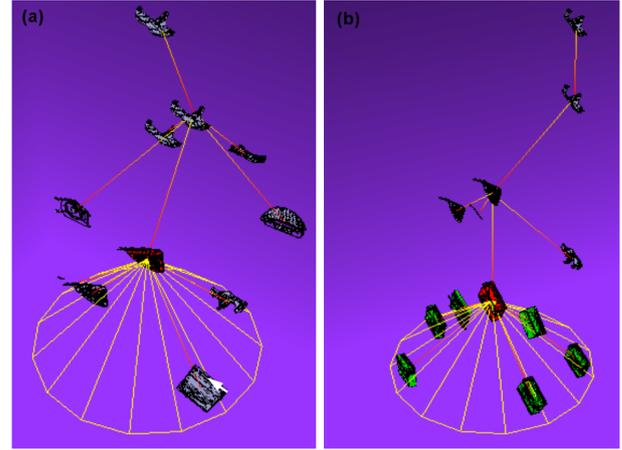


Figure 5 – Illustration du fonctionnement de l'interface, l'utilisateur sélectionne un objet (a), celui-ci devient rouge et son sous-groupe est affiché (b).

seul désavantage est que la clarté souhaitée n'est réalisable qu'au prix d'une perte d'information.

## 6 Résultats

Dans le cadre du projet SEMANTIC-3D, nous disposons de deux bases de 116 et de 758 d'objets 3D organisée en 25 et 75 classes. Le fait de connaître les groupes effectifs nous a permis de pouvoir vérifier la qualité de notre partitionnement. Ainsi, nous avons défini le pourcentage d'objets bien classés comme suit :

$$P(E) = \frac{\sum_{i=0}^G \max_{j \in [0..k]} C(P_i(E_j))}{N} \quad (4)$$

avec  $E$  la  $k$ -partition trouvée par l'algorithme testé,  $C(P_i(E_j))$  le nombre de points du groupe réel  $i$  présents dans le groupe  $E_j$ ,  $G$  le nombre de groupes réels et  $N$  le nombre d'objets du jeu d'essai considéré.

Ce critère favorise les partitions qui conserve le maximum de groupes entiers. Par exemple, si l'algorithme trouve un seul groupe contenant tous les objets, le pourcentage d'objets bien classés sera de 100. Pour contrebalancer ceci, nous affichons également le nombre de groupes trouvés par l'algorithme et le nombre de groupes réels qui ne sont pas fragmentés (i.e les groupes entiers). Le tableau 1 présente les résultats obtenus pour des jeux d'essais de 116 et 758 points.

Les tests ont été effectués sur un Pentium IV 2,66GHz avec 512 Mo de Ram. On constate que notre algorithme donne des résultats sensiblement meilleurs pour des temps de calculs nettement inférieurs. Le fait que le temps d'exécution de notre algorithme suivie des nuées dynamiques floues soit parfois inférieur au temps d'exécution des nuées dynamiques floues seules peut être expliqué par le fait que les centres de groupes soient déjà placés de façon quasi optimale.

Base de 116 objets	Q(%)	$N_G$	$N_E$	T(sec.)
Notre algorithme	91,37	24	18/25	0,19
Notre algo+NDF	92,24	24	19/25	1,59
NDF	89,36	25	17/25	1,07

Base de 758 objets	Q(%)	$N_G$	$N_E$	T(sec.)
Notre algorithme	77,17	81	43/75	3,55
Notre algo+NDF	80,47	81	49/75	78,41
NDF	74,23	75	45/75	156,5

Tableau 1 – Résultats comparés pour notre algorithme et les nuées dynamiques floues (NDF) obtenus sur deux jeux d'essais de 116 et de 758 objets répartis en 25 et 75 classes, en terme de : qualité de partitionnement  $Q$ , nombre de groupe  $N_G$ , nombre de groupes entiers  $N_E$  et temps d'exécution  $T$ .

## 7 Conclusion

Nous avons présentés un système de navigation offrant un algorithme de partitionnement efficace et très rapide. La possibilité de le coupler avec les nuées dynamiques floues permet d'obtenir une structure souple et plusieurs chemins existants pour accéder à un objet. Le système de visualisation 3D proposé présente plusieurs avantages : outre sa clarté et sa simplicité d'utilisation, la représentation du chemin suivi dans l'arborescence permet à l'utilisateur de mieux appréhender l'organisation de la base et donc de la parcourir plus rapidement. Les extensions possibles de ce système sont nombreuses : l'utilisation d'autres jeux de descripteurs ou la possibilité d'établir la correspondance, à toutes les étapes du processus de navigation, entre les objets 3D et des vues 2D de ces objets en sont des exemples. Mais, prioritairement, notre intention est de mettre à profit la rapidité de l'algorithme de classification pour permettre à l'utilisateur d'interagir avec le système en modifiant l'agencement des objets en cours de navigation.

## Remerciement

Ce travail est financé par le Ministère Français de la recherche dans le cadre du projet RNRT SEMANTIC-3D (<http://www.semantic-3d.net>).

## Références

- [1] S. Craver, B. Yeo, et M. Yeung. Multi-linearization data structure for image browsing. Dans *SPIE Storage and Retrieval of Image and Video Databases (SPIE '99)*, pages 155–166, San Jose, CA, USA, Janvier 1999.
- [2] Jau-Yuen Chen, Charles A. Bouman, et John C. Dalton. Hierarchical browsing and search of large image databases. *IEEETIP : IEEE Transactions on Image Processing*, 9(3), 2000.
- [3] Teuvo Kohonen. *Self-organizing maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.
- [4] J. Bezdek. *Pattern Recognition with Fuzzy Objective Functions Algorithms*. Plenum, New York, USA, 1981.
- [5] H. Frigui et R. Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition Journal*, 30(7) :1109–1119, 1997.
- [6] Bertrand Le Saux et Nozha Boujemaa. Image database clustering with svm-based class personalization. Dans *Conference on Storage and Retrieval Methods and Applications for Multimedia / Electronic Imaging symposium (SPIE '04)*, San Jose, CA, USA, Janvier 2004.
- [7] Daniel Boley. Principal direction divisive partitioning. *Data Min. Knowl. Discov.*, 2(4) :325–344, 1998.
- [8] B. Moghaddam, Q. Tian, et T. S. Huang. Spatial visualization for content-based image retrieval. Dans *IEEE International Conference on Multimedia and Expo (ICME01)*, Tokyo, Japan, Août 2001.
- [9] Simone Santini, Amarnath Gupta, et Ramesh Jain. Emergent semantics through interaction in image databases. *IEEE Transactions on Knowledge and Data Engineering*, 13(3) :337–351, 2001.
- [10] K. Idrissi, G. Lavoue, J. Ricard, et A. Baskurt. Object of interest based visual navigation, retrieval and semantic content identification system special issue on colour for image indexing and retrieval. *Computer Vision On Image Understanding*, 94(1-3) :271–294, 2004.
- [11] George G. Robertson, Jock D. Mackinlay, et Stuart K. Card. Cone trees : animated 3d visualizations of hierarchical information. Dans *SIGCHI conference on Human factors in computing systems*, pages 189–194, New York, NY, USA, 1991. ACM Press.
- [12] W. J. Rogers, S. J. Cunningham, et G. Holmes. Navigating the virtual library : a 3d browsing interface for information retrieval. *Australian Journal of Intelligent Information Processing Systems*, 3(4) :43–49, 1996.
- [13] C. Chen et S. Morris. Visualizing evolving networks : Minimum spanning trees versus pathfinder networks. Dans *IEEE Symposium on Information Visualization*, pages 67–74, Seattle, Washington, 2003, 2003.
- [14] Julien Ricard, David Coeurjolly, et Atilla Baskurt. Generalizations of angular radial transform for 2d and 3d shape retrieval. *Pattern Recognition Letters*, accepted for publication, 2005.
- [15] Christos Faloutsos et King-Ip Lin. Fastmap : a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. Dans *ACM SIGMOD international conference on Management of data*, pages 163–174, New York, NY, USA, 1995. ACM Press.