

ATELIER FOUILLE VISUELLE DE DONNEES : AVANCEES RECENTES ET PERSPECTIVES

Dans le cadre de la conférence Extraction et Gestion de Connaissances
(EGC'2011)

Brest, 25 janvier 2010

Hanene Azzag, Fabien Picarougne, Gilles Venturini

Atelier Fouille Visuelle de Données : avancées récentes et perspectives

Hanene Azzag*, Fabien Picarougne**
Gilles Venturini***

*Université Paris 13, LIPN-UMR 7030 - CNRS 99, avenue Jean-Baptiste Clément
93430 Villetaneuse

hanene.azzag@lipn.univ-paris13.fr,
<http://www-lipn.univ-paris13.fr/azzag/>

** LINA - Polytech'Nantes, rue Christian Pauc
BP50609 F-44306 Nantes Cedex 3
fabien.picarougne@univ-nantes.fr,
<http://www.polytech.univ-nantes.fr/COD/>

*** LI - Polytech'Tours, 64 avenue Jean Portalis
37200 Tours
venturini@univ-tours.fr,
<http://www.antsearch.univ-tours.fr>

Cette première édition de l'atelier « Fouille Visuelle de Données : avancées récentes et perspectives » se déroulant dans le cadre d'EGC 2011 a pour but de réunir les chercheurs intéressés par les approches visuelles et interactives en Fouille de Données et en Extraction de Connaissances. Notre ambition est de permettre aux participants d'aborder tous les thèmes de la Fouille Visuelle de Données et des interactions entre l'expert du domaine et les outils de fouille. L'atelier concerne aussi bien les chercheurs du monde académique que ceux du secteur industriel, et autant les notions conceptuelles que les applications grâce à du matériel spécifique permettant la réalisation de démonstrations de visualisations en 2D et en 3D stéréoscopique.

Les thèmes abordés en Fouille Visuelle de Données sont très nombreux et nous nous intéresserons principalement aux questions relatives à la combinaison des approches visuelles et des approches classiques, à l'apport de la 3D, à l'évaluation et notamment à la spécificité de l'évaluation en Fouille Visuelle de Données. Nous débattons également sur les succès et échecs récemment rencontrés.

Comité d'organisation

- Hanene Azzag (LIPN, Université de Paris 13)
- Fabien Picarougne (LINA, Université de Nantes)
- Gilles Venturini (LI, Université François Rabelais de Tours)

Comité de programme

- Bruno Arnaldi (IRISA, INSA de Rennes)
- Khalid Benabdeslem (LIRIS, Université Lyon I)
- Younès Bennani (LIPN, Université de Paris 13)
- Fatma Bouali (Université de Lille 2)
- Guillaume Cleuziou (LIFO, Université d'Orléans)
- Mathieu Exbrayat (LIFO, Université d'Orléans)
- Nazim Fates (INRIA Nancy Grand-Est)
- Stéphane Huot (LRI, Université de Paris-Sud)
- Jean-Pierre Jessel (IRIT, Université Toulouse 3)
- Mustapha Lebbah (LIPN, Université de Paris 13)
- Lionel Martin, (LIFO, Université d'Orléans)
- Guy Mélançon (LABRI, Université de Bordeaux)
- Guillaume Moreau (CERMA, Ecole centrale de Nantes)
- Monique Noirhomme (Institut d'Informatique, FUNDP, Namur, Belgique)
- Benoit Otjacques, (Centre Gabriel Lippmann, Luxembourg)
- Fabrice Rossi (Télécom ParisTech)

Programme

14h - 14h30

Une nouvelle approche anthropocentrée de visualisation d'une couverture optimale de règles d'association guidée par la pertinence cognitive

Amira Mouakher, Sadok Ben Yahia

14h30 - 15h

Exploration visuelle de la classification des données mixtes séquentielles

Rakia Jaziri, Mustapha Lebbah, Younès Bennani, Jean-Hugues Chenot

15h - 15h30

Visualisation des SVM et des changements de représentations induits par les noyaux

Matthieu Exbrayat, Lionel Martin

15h30 - 16h

Pause café

16h - 16h30

DÉMONSTRATION : Tulip : a Scalable and Adaptive Graph Visualization Framework

David Auber, Romain Bourqui, Maylis Delest, Jonathan Dubois, Ludwig Fiolka, Charles Huet, Antoine Lambert, Patrick Mary, Morgan Mathiaut, Guy Melançon, Bruno Pinaud

16h30-17h

Acquisition, Visualisation 3D et interaction pour le suivi de dissection de fibres cérébrales du cerveau humain

Barthélemy Serres, Christophe Destrieux, Gilles Venturini

17h-17h30

Librairie Open Source de Visualisation Interactive et immersive

Nicolas Greffard, Fabien Picarougne, Pascale Kuntz

Une nouvelle approche anthropocentrée de visualisation d'une couverture optimale de règles d'association guidée par la pertinence cognitive

A. MOUAKHER et S. BEN YAHIA

Faculté des Sciences de Tunis
sadok.benyahia@fst.rnu.tn

Résumé. La fouille visuelle des données est un nouveau domaine qui a donné naissance à plusieurs approches où les capacités visuelles sont exploitées pour extraire des connaissances utiles. À cet égard, l'implication de l'utilisateur au coeur du processus d'extraction semble être un facteur clé dans la réussite de tout système de visualisation. Dans cet article, nous introduisons une nouvelle approche anthropocentrée de visualisation d'une couverture optimale de règles d'association. Étant donné que l'extraction d'une telle couverture est un problème NP complet, nous présentons un algorithme glouton OPTCOVER qui se base sur la pertinence cognitive des règles lors de la construction de la couverture de concepts formels. En outre, nous présentons une description de quelques interfaces illustrant les principales caractéristiques de notre système de visualisation qui repose sur le paradigme de la réalité virtuelle.

1 Introduction et motivations

Avec l'accroissement considérable de la quantité d'information, nous faisons face à beaucoup de difficultés, essentiellement, durant la phase d'interprétation de connaissances extraites à partir de grandes masses de données. Plusieurs solutions ont été proposées pour remédier à ce problème : Lors de la première décennie, les recherches se sont essentiellement focalisées sur le développement d'algorithmes automatiques performants. Cependant, durant ces dernières années, l'implication de l'utilisateur au sein du processus d'extraction est peu à peu devenu un sujet de préoccupation majeur Kuntz et al. (2006). Ce besoin d'intégration a conduit à l'émergence de nombreux outils de visualisation et d'interaction Fayyad et al. (2001).

Le reste du papier est organisé de la manière suivante. Dans la section 2, nous commençons par présenter un survol de l'état de l'art sur les techniques de représentation visuelles des règles d'association. Ensuite, nous décrivons dans la section 3, notre nouvelle approche composée de deux phases à savoir, une phase d'extraction de couverture et une phase de visualisation L'étude expérimentale de notre approche est illustrée dans la section 4. Nous terminons l'article par une conclusion ainsi que les travaux en cours.

2 Visualisation des règles d'association : état de l'art

La fouille visuelle de données ("*Visual Data Mining*") joue actuellement un rôle important dans le processus d'extraction de connaissances. Elle permet non seulement d'aider l'expert à

valider des hypothèses sur les données, mais également de suivre et de comprendre le comportement des algorithmes d'exploration (de Oliveira et Levkowitz (2003)). À cet effet, plusieurs représentations visuelles ont été proposées dans la littérature pour assurer la visualisation post la phase d'extraction aussi bien qu'intra phase d'extraction. Dans ce qui suit, nous allons passer en revue une synthèse des travaux antérieurs sur la visualisation des règles d'association.

2.1 Visualisation post phase d'extraction

Comme il a été mentionné par C.H.Yamamoto et al. (2009), les représentations les mieux adaptées pour afficher les règles d'association après la phase d'extraction sont sous forme de grilles et d'histogrammes.

Blanchard et al. (2003b) ont proposé un prototype expérimental, appelé *ArVis*, qui a pour objectif d'attaquer le problème de validation des règles, par la proposition d'une approche anthropocentrée pour la tâche d'exploration de règles. L'approche proposée, basée sur un modèle navigationnel spécifique, repose sur les mesures d'intérêt des règles reliées. Ce travail a été étendu dans Blanchard et al. (2007b) avec deux types de visualisation complémentaires : la première est la métaphore visuelle 3D et la seconde est une visualisation 2D des jeux de règles et de leurs relations de voisinage. Dans une autre extension (Blanchard et al. (2007a)), les relations de voisinage ont été ajoutées à l'interface *ARVIS* avec leurs contraintes syntaxiques et ceux des mesures d'intérêt.

Par ailleurs, Couturier et al. (2005) ont proposé une approche basée sur une fouille hiérarchique de règles d'association, en utilisant une approche guidée par l'utilisateur au lieu d'une approche automatique. Dans cette approche anthropocentrique, l'utilisateur constitue le coeur du processus en jouant le rôle d'une heuristique évolutionnaire.

D'autres auteurs ont combiné plusieurs représentations interactives des règles d'association avec la technique de visualisation de type *Fisheye* (oeil de poisson). L'approche de Couturier et al. (2007) rentre dans ce cadre et permet de produire des vues en deux dimensions où les structures visuelles de la représentation apparaissent avec des tailles variables. L'objectif majeur de cette technique est de donner aux utilisateurs les détails à la demande tout en préservant un aperçu du contexte global. Les mêmes auteurs (Couturier et al. (2007)) ont proposé un outil appelé *CbVAR* pour explorer des règles d'association fondée sur une représentation visuelle des clusters de règles, et ceci en appliquant l'algorithme k-means sur les caractéristiques des règles.

2.2 Visualisation intra phase d'extraction

Il est vrai que les approches qui visent la visualisation après la phase d'extraction permettent à l'utilisateur une meilleure perception des règles pertinentes dans le corpus des règles inférées, néanmoins, elles n'encouragent pas son implication au coeur du processus d'extraction. Le travail de Zaki et Phoophakdee (2003), introduisant l'outil *MIRAGE*, présente une nouvelle approche permettant d'extraire et d'explorer visuellement des règles *minimales*. *MIRAGE* utilise une approche de visualisation de règles interactive basée sur l'exploration du treillis de Galois, permettant d'afficher les règles dans une forme très compacte. Yang (2003) a employé les coordonnées parallèles pour visualiser les itemsets fréquents et les règles d'association. Initialement, les items sont rangés par groupes de sorte que ceux de même cardinalité apparaissent dans le même voisinage. Ensuite, n axes sont dessinés de façon à ce que n soit

le nombre d'items présents dans l'itemset le plus large. Les itemsets fréquents et les règles d'association sont représentés par des traits discontinus reliant ces axes. Mahanti et Alhajj (2005) ont développé une interface pour afficher le treillis des motifs fréquents obtenus lors de l'extraction des règles. Cette représentation, bien qu'elle soit orientée vers l'apprentissage plutôt que la génération des règles, est très utile pour les experts puisqu'elle fournit un aperçu sur le processus de génération de motifs fréquents. Liu et Salvendy (2006) ont proposé une technique permettant la visualisation des itemsets fréquents ainsi que les règles d'association extraites à chaque itération de l'algorithme APRIORI. Elle est basée sur une structure grille qui affiche les items et les itemsets respectivement sur les lignes et les colonnes. Bien que cette approche donnait la main à l'utilisateur pour supprimer les itemsets inutiles, cette dernière ne garantit pas une interaction totale avec les données. En effet, l'utilisateur n'est pas autorisé à faire un retour arrière une fois que l'exécution du processus d'extraction est lancée. Dans ce qui suit, nous allons introduire une nouvelle approche de fouille visuelle anthropocentrée de règles d'association.

3 Nouvelle approche de fouille visuelle anthropocentrée

Dans cette section, nous décrivons une nouvelle approche pour l'extraction d'une couverture optimale guidée par la pertinence cognitive. En effet, dans notre approche, le processus d'extraction de règles se décompose en deux phases : dans un premier lieu, nous procédons à la réduction de la charge cognitive de l'utilisateur à travers un algorithme d'extraction d'une couverture pertinente sans perte d'information. Ensuite, l'ensemble de règles d'association est généré à partir de cette couverture et la tâche de visualisation est ainsi fortement simplifiée pour l'utilisateur.

3.1 Extraction de couverture optimale d'une relation binaire

Dans ce qui suit, nous allons commencer par passer en revue les travaux les plus saillants sur l'extraction d'une couverture optimale.

3.1.1 Travaux antérieurs sur l'extraction de couverture

La construction d'une couverture optimale d'une relation binaire \mathcal{R} est un problème connu comme NP complet (Garey et Johnson (1979)). Les algorithmes exacts qui lui ont été proposés sont donc, de complexité exponentielle. Ainsi, le recours à des solutions heuristiques s'imposait comme une solution *Sine qua non*. Belkhit et al. (1994) s'inspirent de la méthode de séparation et d'évaluation progressive appelée aussi, selon Brassard et Bratley (1987), "*Branch and Bound*" et proposent une heuristique qui consiste à évaluer chaque concept $C = \langle E, I \rangle$ selon une fonction de gain donnée par l'équation 1 et à retenir celui qui la maximise.

$$gain(C) = (|E| \times |I|) - (|E| + |I|) \quad (1)$$

Plus tard, Khchérif et al. (2000) ont introduit un algorithme qui se base sur un calcul algébrique d'une relation particulière dite *difonctionnelle* Riguet (1944, 1998) donnée par l'équation 2. Cette difonctionnelle permet d'identifier les couples qui ne sont couverts que par un seul concept, et construit ces concepts.

$$\mathcal{R}^d = \overline{\mathcal{R} \circ \mathcal{R}^{-1} \circ \mathcal{R}} \cap \mathcal{R} \quad (2)$$

où $\overline{\mathcal{R}}$ est la relation complément de \mathcal{R} et \mathcal{R}^{-1} est la relation inverse de \mathcal{R} .

Récemment, Belohlavek et Vychodil (2009) ont abordé le même problème en proposant une nouvelle méthode de décomposition de matrices binaires. Ils prouvent un théorème d'optimalité, qui affirme que la plus petite couverture qui puisse représenter le contexte est construite à partir de concepts formels. La principale critique qui peut être assignée à ces approches, est que la couverture pertinente est évaluée à travers des heuristiques statiques qui ne reflètent forcément pas la qualité de connaissances qui peuvent en être extraite. C'est pour cette raison que dans notre approche, la fonction de gain est basée sur des mesures de corrélation.

3.1.2 Description de l'algorithme OPTCOVER

La principale originalité de cet algorithme glouton, que nous introduisons, réside dans l'association de la pertinence d'un concept formel avec une mesure de corrélation des règles d'association qu'il va générer. Parmi ces mesures nous pouvons citer *la confiance*, *la mesure lift*, *la mesure any-confidence*, *la mesure all-confidence*, *la mesure bond*, etc. Pour plus d'informations sur ces mesures, le lecteur peut se référer à l'article de Omiecinski (2003).

Cette approche ne calcule pas tout le treillis de concepts. Elle se limite plutôt à chercher un sous-ensemble de concepts pertinents, appelée *couverture de concepts* à partir de laquelle nous pouvons induire les règles d'association. Ces concepts remplacent tout le treillis sans perte et sans redondance d'information. En fait, le treillis peut être recalculé à partir de la couverture en utilisant les opérateurs relationnels de supremum et d'infimum.

OPTCOVER prend en entrée un contexte formel \mathcal{K} , muni d'une mesure de corrélation paramétrée H , choisie préalablement par l'utilisateur ainsi qu'un seuil minimal de corrélation min_H et d'un seuil minimal de confiance σ_{cf} pour la génération des règles d'association. Il retourne une couverture minimale de concepts pertinents \mathcal{F}_c ainsi que l'ensemble de règles d'association \mathcal{R}_c correspondant. Cette construction est assistée par l'utilisateur qui peut exprimer ses préférences à travers la validation ou le rejet des choix de l'algorithme.

3.1.3 Pseudo-code de l'algorithme

Le pseudo-code de l'algorithme est récapitulé par l'algorithme 1 : soit φ l'ensemble des couples (o, i) à explorer. Cette variable contient au début de l'algorithme tous les couples du contexte \mathcal{K} (ligne 2). Étant donné que notre démarche est essentiellement incrémentale, la couverture de concepts pertinents \mathcal{F}_c ainsi que l'ensemble de règles générées \mathcal{R}_c sont initialisés à l'ensemble vide au début de l'algorithme (ligne 3-4).

Après avoir calculé les pseudo-concepts de tous les couples appartenant à φ (ligne 6), la fonction CALCUL-GAIN fournit le gain ω relatif à chaque pseudo-concept $PCF_{(o,i)}$ selon la mesure de corrélation H choisie (ligne 7).

Un tri est alors effectué sur les couples par ordre décroissant de pertinence (ligne 8). Ensuite, pour chacun des couples (o, i) pris dans cet ordre, nous générons la liste des concepts S_c contenus dans le pseudo-concept correspondant $PCF_{(o,i)}$ à travers la fonction GÉNÉRER-CONCEPTS (ligne 10). La dérivation des règles d'association est effectuée par la fonction DÉRIVER-RÈGLES-CONCEPTS et l'algorithme peut ainsi déterminer le concept pertinent $\langle A, B \rangle$ grâce à la fonction CALCUL-CONCEPT-PERTINENT (ligne 12). Ce dernier est alors ajouté à la

liste des concepts pertinents \mathcal{F}_c et l'ensemble des règles \mathcal{R}_c est aussi mis à jour (ligne 13-14). Avant de passer à l'itération suivante, nous procédons à l'élimination des couples redondants et au retrait de tous les couples couverts par le concept retenu (ligne 16). L'algorithme se termine lorsque l'ensemble de couples à ajouter, φ , est épuisé et retourne l'ensemble des concepts pertinents \mathcal{F}_c et les règles d'association correspondantes \mathcal{R}_c (ligne 17).

Données :

1. $\mathcal{K}=(\mathcal{O}, \mathcal{I}, \mathcal{R})$: contexte formel.
2. H : mesure de corrélation.
3. min_H : seuil minimal de corrélation.
4. σ_{cf} : seuil minimal de confiance.

Résultats :

1. \mathcal{F}_c : couverture de concepts pertinents.
2. \mathcal{R}_c : ensemble de règles d'association extraites à partir de la couverture de concepts pertinents.

```

1  début
2  |  $\varphi = \{(o, i) \in \mathcal{R}\}$ ;
3  |  $\mathcal{F}_c = \emptyset$ ;
4  |  $\mathcal{R}_c = \emptyset$ ;
5  | pour chaque  $(o, i) \in \varphi$  faire
6  | |  $PCF_{(o,i)} = \text{CALCUL-PSEUDOCONCEPT}(\mathcal{K}, o, i)$ ;
7  | |  $\omega = \text{CALCUL-GAIN}(\mathcal{K}, PCF_{(o,i)}, H)$ ;
8  | |  $\text{TRIER-ELEMENTS}(\varphi, \omega)$ ;
9  | | tant que  $(\varphi \text{ est non vide})$  faire
10 | | |  $S_c = \text{GÉNÉRER-CONCEPTS}(\mathcal{K}, PCF_{(o,i)})$ ;
11 | | |  $\mathcal{R} = \text{DÉRIVER-RÈGLES-CONCEPTS}(S_c, \sigma_{cf})$ ;
12 | | |  $\langle A, B \rangle = \text{CALCUL-CONCEPT-PERTINENT}(\mathcal{K}, S_c, \mathcal{R}, H, min_H, \mathcal{R}_{A,B})$ ;
13 | | |  $\mathcal{F}_c = \mathcal{F}_c \cup \langle A, B \rangle$ ;
14 | | |  $\mathcal{R}_c = \mathcal{R}_c \cup \mathcal{R}_{A,B}$ ;
15 | | | pour chaque  $(a, b) \in A \times B$  faire
16 | | | |  $\varphi = \varphi - (a, b)$ ;
17 | retourner  $(\mathcal{F}_c, \mathcal{R}_c)$ 
18 fin

```

Algorithme 1 : ALGORITHME OPTCOVER

3.2 Visualisation de la couverture optimale

Dans ce qui suit, nous allons justifier dans un premier lieu notre choix d'un modèle anthropocentré pour la fouille de règles d'association. Ensuite, nous allons expliquer la métaphore visuelle ainsi que le modèle cognitif adopté dans notre approche de visualisation. Finalement, nous allons mettre en exergue le rôle interactif de l'utilisateur avec notre système de visualisation.

3.2.1 Besoin d'un modèle anthropocentré

L'évaluation de la pertinence des règles d'association ne peut pas se faire de façon absolue puisqu'elle dépend essentiellement des préférences, des objectifs de l'utilisateur ainsi que les mesures de qualité choisies pour un contexte donné. L'utilisateur a donc besoin d'explorer les données pour pouvoir interpréter et identifier les sous-ensembles qu'il juge "intéressantes". Étant donnée que cette exploration constitue un processus non supervisé et vue l'ampleur des règles d'association issues des algorithmes d'extraction, l'utilisateur se trouve face à une tâche difficile à accomplir Blanchard et al. (2003a). Afin de remédier à ces handicaps, nous avons adopté dans notre approche de visualisation un modèle anthropocentré dans lequel l'utilisateur joue le rôle d'une heuristique qui dirige le processus d'extraction et exploite son analyse pour affiner les résultats.

3.2.2 Métaphore visuelle

Pour générer la représentation visuelle des règles d'association, nous avons adopté une représentation 3D des mondes virtuels appelée aussi *cyber-espaces* (Chen (2004)). Avec ce type de visualisation, les données sont représentées par des objets répartis dans un grand espace affiché en 3D au sein duquel l'utilisateur peut librement naviguer et contrôler son point de vue Blanchard (2005). De ce fait, on présume que l'interaction avec l'utilisateur se fait essentiellement via la navigation.

Dans notre approche de visualisation, nous avons adopté la métaphore du *paysage d'information* (Andrews (1995); Robertson et al. (1998)) qui présente l'un des types de la représentation des mondes virtuels. Ce choix a été établi afin d'optimiser la visibilité des objets et de diminuer les occultations. Pour ne pas encombrer l'interface de visualisation, les règles d'association ne sont affichées qu'à la demande. En effet, au début de la visite du paysage 3D, l'utilisateur est placé au centre de la scène et ne visualise que les concepts formels générés à chaque itération.

Nous avons choisi de symboliser chaque concept formel par l'objet suivant : un cône posé au dessous d'une sphère. L'encodage graphique illustré par la figure 1 est comme suit :

- La position de l'objet est relative à la valeur du gain du concept : on classe les concepts par ordre de gain décroissant. Ainsi, plus le concept est pertinent moins il est éloigné dans notre environnement virtuel.
- La couleur du concept pertinent est différente des autres.
- Le diamètre de la base du cône indique le pourcentage du nombre de règles générées à partir du concept.
- La hauteur du cône est proportionnelle au gain associé au concept.
- Le diamètre de la sphère représente le pourcentage du gain par rapport aux restes des concepts.

En ce qui concerne l'ensemble des règles d'association, nous avons choisi de les représenter par groupe. Plus précisément, l'utilisateur ne visualise que les règles générées lors de l'itération courante..

L'encodage graphique illustré par la figure 1 est le suivant :

- Chaque sous-ensemble de règles est placé dans la même région du concept à partir duquel il est généré.

- Chaque règle est représentée par un histogramme 3D placé sur une grille orientée par un repère à 3 axes : un premier axe pour les prémisses, un deuxième axe pour les conclusions et un troisième axe pour les valeurs de gain relatifs aux règles.
- La hauteur de l'histogramme représente la valeur du gain de la règle qui est aussi affichée comme un label pour qu'elle soit repérée par l'utilisateur.

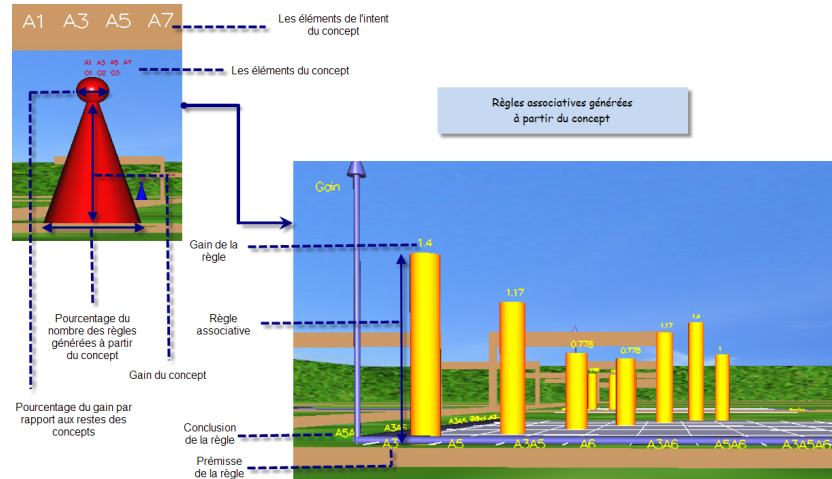


FIG. 1 – Encodage graphique des concepts et des règles d'association

3.2.3 Modèle cognitif de visualisation

Nous avons représenté le modèle cognitif adopté pour notre système de visualisation à travers la figure 2. Ainsi, nous pouvons distinguer que l'utilisateur constitue le noyau de notre système de visualisation : il fixe les paramètres de l'algorithme d'extraction et effectue des choix intermédiaires qui vont orienter la suite du processus. Afin d'améliorer l'intelligibilité de la représentation, notre système ne fournit pas à l'utilisateur la totalité des règles extraites en une seule passe. Il procède par étape et n'affiche donc que les règles générées à partir des concepts formels de l'itération courante. Il permet aussi une visualisation des résultats sans intervention de l'utilisateur.

3.2.4 Interactions avec l'utilisateur

Comme cela a été mis en exergue par Hascoët et Beaudouin-Lafon (2001), la visualisation se doit d'être interactive afin d'être réellement efficace. C'est pour cette raison que notre système de visualisation se base essentiellement sur l'interaction avec l'utilisateur. En effet, cette interaction se manifeste par deux manières différentes :

- Une première manière consiste à agir directement sur la représentation et ceci s'effectue par la navigation de l'utilisateur dans la scène pour pouvoir cibler les données intéressantes lors de l'exploration des concepts générés par notre algorithme d'extraction. Le

Une nouvelle approche anthropocentrée de visualisation

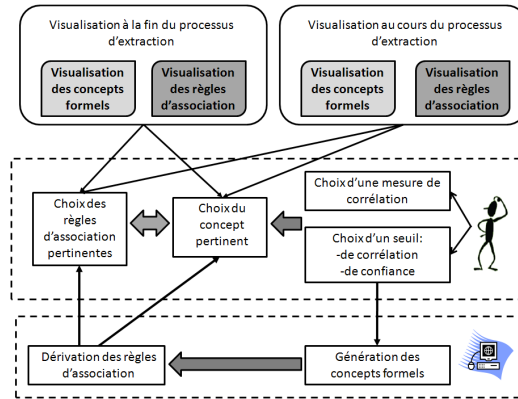


FIG. 2 – *Modèle de visualisation*

mode de navigation est fixé préalablement par l'utilisateur (clavier, souris, gant, wii-mote), qui est initialement placé au centre de la scène pour bénéficier d'un point de vue synthétique. Il peut ensuite se promener librement dans le paysage 3D pour examiner les règles de plus près et assurer une meilleure observation en modifiant la position de la caméra.

- Une deuxième manière d'interaction se concrétise lors du processus d'extraction lui-même. En effet, l'utilisateur est amené soit à valider le choix recommandé par l'algorithme, soit à faire des choix différents. Son choix est déterminé à travers la visualisation des règles associatives extraites lors de chaque itération. Il peut donc en choisir celles qui lui semblent les plus pertinentes. En outre, l'utilisateur a la possibilité de revenir sur ses décisions ou sauter toutes les itérations en une seule passe. Cette interaction est possible à travers une barre de contrôle qui est placée au dessous de la scène de visualisation.

Description de l'interface de visualisation

L'interface principale de notre outil de visualisation est illustrée par la figure 3(Gauche). Comme nous pouvons le remarquer, elle est divisée en trois espaces :

1. **Espace de paramétrage** : Avant de procéder au processus d'extraction, l'utilisateur est amené à fixer certains paramètres, e.g., le choix du fichier à analyser, les seuils ainsi que le mode de déplacement (clavier, souris, gant, wiimote).
2. **Espace de contrôle** : À travers cette barre de contrôle, l'utilisateur est impliqué au cœur du processus d'extraction. En effet, ce dernier a la possibilité d'exprimer ses choix par la validation ou le rejet des choix de l'algorithme. Il peut aussi revenir sur ces décisions ou demander au prototype de dérouler l'algorithme sans son intervention.
3. **Espace de visualisation** : Cet espace représente la scène de visualisation. Elle est sous forme d'une arène cubique pour faciliter la perception de la profondeur dans l'environnement 3D. À chaque itération, cette scène est dynamiquement divisée en régions selon le nombre de concepts générés qui sont disposés par ordre décroissant de pertinence.

Chaque concept est identifié par un drapeau indiquant les éléments de son intent à partir desquels on va extraire les règles d'association. En outre, des étiquettes textuelles complémentaires sont affichées au-dessus de chaque concept pour donner les objets et les attributs qui le constituent.

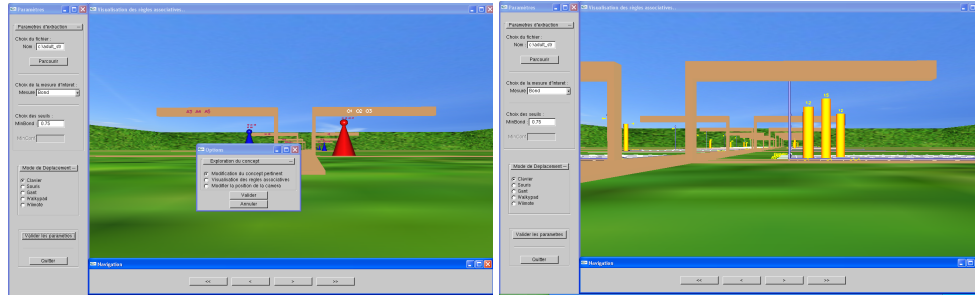


FIG. 3 – (**Gauche**) Concept formel représenté sous la forme d'un cône (**Droite**) Navigation au sein de l'ensemble des règles d'association

Une fois les paramètres de l'algorithme sont fixées, l'utilisateur peut contrôler le processus d'extraction. Afin de faciliter la tâche d'exploration, notre outil de visualisation permet à l'utilisateur non seulement d'avoir une idée synthétique sur la pertinence des concepts mais aussi une vue plus détaillée comme illustré dans la figure 3 (Droite) au cas où il souhaite examiner les règles d'association de près et ceci en double cliquant sur le concept.

4 Étude expérimentale

Notre étude expérimentale portera sur l'aspect d'extraction de couverture optimale. Nous allons donc évaluer notre algorithme OPTCOVER en appliquant les différentes mesures de corrélation dans un premier lieu et en se comparant avec les approches de la littérature dans un second lieu.

Nous avons comparé, à travers une étude expérimentale approfondie, la taille de notre couverture optimale vs le nombre de concepts formels contenus dans des bases benchmark ⁽¹⁾. Les caractéristiques de ces bases sont résumées par le tableau 1. Ce tableau définit, pour chaque base, le nombre d'attributs, le nombre d'objets, le nombre de concepts formels ainsi que les tailles des couvertures générées avec les mesures de corrélation Lift, Confiance et Bond. Pour toutes ces expérimentations, la valeur de *minconf* est fixée à 1 et les seuils de corrélation sont égaux à 0.75. En accord avec le théorème "No Free Lunch Theorem" Wolpert et Macready (1997), aucune mesure d'intérêt ne peut être considérée comme la meilleure en terme de compacité. Cependant, comme nous pouvons le remarquer dans la table 1, la couverture basée sur la mesure Bond donne dans la plupart des cas la couverture la plus compacte.

Comme nous pouvons le constater à travers le tableau 2, la couverture basée sur la mesure *Bond* est la plus réduite en terme de compacité pour toutes les bases considérées. Néanmoins, cette couverture ne fournit pas les règles d'association intéressantes de point de vue cognitif. Des résultats similaires ont été obtenues par la couverture de Belohlavek et Vychodil (2009).

1. Ces bases sont fournies par UC Irvine Machine Learning Database Repository : <http://archive.ics.uci.edu/ml/>.

Base	# attributs	# objets	#concepts formels	#(Lc)	#(Cc)	#(Bc)
SHUTTLE- LANDING- CONTROL	15	24	52	18	23	19
ADULT- STRETCH	20	10	89	14	16	9
LENSES	24	12	128	23	32	12
ZOO	101	28	377	52	40	25
HAYES-ROTH	132	18	380	53	45	17
SERVO	167	19	432	59	72	19
POST- OPERATIVE	90	25	1521	60	55	22

TAB. 1 – Caractéristiques des bases benchmark et les différentes couvertures obtenues. (Lc) : Couverture de Lift , (Cc) : Couverture de la Confiance, (Bc) : Couverture de Bond.

Base	#concepts formels	#Couv.Bond	#Couv. Belkhiter	#Couv. Belohlavek
SHUTTLE- LANDING- CONTROL	52	19	20	22
ADULT- STRETCH	89	9	14	10
LENSES	128	12	21	12
ZOO	377	25	43	26
HAYES-ROTH	380	17	49	17
SERVO	432	19	60	19
POST- OPERATIVE	1521	22	52	22

TAB. 2 – Comparaison avec les approches de littérature en terme de compacité

Actuellement, nous cherchons à mener une évaluation de l'utilisation de notre système de visualisation. Le but serait d'estimer l'apport de la visualisation 3D-stéréo. Ainsi, un questionnaire d'évaluation devrait être mis en place et rempli par des utilisateurs ayant des connaissances variables en Informatique et en Fouille de données.

5 Conclusion et perspectives

Dans ce papier, nous avons proposé une nouvelle approche de visualisation de règles d'association. En effet, cette dernière présente l'avantage d'être orientée utilisateur, et par ce fait, ce dernier est fortement impliqué dans le processus d'extraction. De plus, notre approche minimise la charge cognitive, sans perte d'information, par la manipulation d'un ensemble réduit de règles d'association extraites à partir d'une couverture optimale de concepts formels. Au delà de l'évaluation de l'interface de visualisation, nous comptons nous intéresser à la visualisation de folksonomies en prenant compte du profil de l'utilisateur. Dans ce cadre, une catégorisation

des différents types des utilisateurs, empruntée du domaine du E-learning, pourrait être mise à profit pour une adéquation visualisation-utilisateur.

Remerciements : Les auteurs souhaitent vivement Pr. Pascale KUNTZ (équipe COD, LINA-Nantes) pour toute son aide fort précieuse en nous donnant accès à l'équipement de visualisation 3D.

Références

- Andrews, K. (1995). Case study : Visualising cyberspace : information visualisation in the harmony internet browser. *IEEE Information Symposium on Visualization 0*, 97.
- Belkhiter, N., C. Bourhfir, M. Gammoudi, A. Jaoua, N. L. Thanh, et M. Reguig (1994). Décomposition Rectangulaire Optimale d'une Relation Binaire : Application aux bases de Données Documentaires. *INFOR 1*, 33–54.
- Belohlavek, R. et V. Vychodil (2009). Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences* 76(1), 3–20.
- Blanchard, J. (2005). *Un système de visualisation pour l'extraction, l'évaluation, et l'exploration interactives des règles d'association*. Ph. D. thesis, École Polytechnique de l'Université de Nantes.
- Blanchard, J., F. Guillet, et H. Briand (2003a). Une visualisation orientée qualité pour la fouille antropocentrée de règles d'association. *Cognito, Cahiers romans de sciences cognitives 1*(3), 79–100.
- Blanchard, J., F. Guillet, et H. Briand (2003b). A user-driven and quality-oriented visualization for mining association rules. In *Proceedings of the Third IEEE International Conference on Data Mining, Melbourne, Florida*, pp. 493–496.
- Blanchard, J., F. Guillet, et sH. Briand (2007a). Interactive visual exploration of association rules with rule-focusing methodology. *Knowl. Inf. Syst. 13*(1), 43–75.
- Blanchard, J., B. Pinaud, P. Kuntz, et F. Guillet (2007b). A 2d-3d visualization support for human-centered rule mining. *Computers & Graphics 31*(3), 350–360.
- Brassard, G. et P. Bratley (1987). *Algorithmique, Conception et Analyse*. Paris : Masson, Paris and Les Presses de l'Université de Montréal.
- Chen, C. (2004). *Information Visualization*. Springer.
- C.H.Yamamoto, M. de Oliveira, et S. Rezende (2009). *Post-Mining of Association Rules : Techniques for Effective Knowledge Extraction*, Chapter Visualization to Assist the Generation and Exploration of Association Rules, pp. 224–245. Information Science Reference.
- Couturier, O., T. Hamrouni, S. B. Yahia, et E. M. Nguifo (2007). A scalable association rule visualization towards displaying large amounts of knowledge. In *Proceedings of 11th International conference on Information Visualization IV'07, Zurich, Switzerland, IEEE Computer Society*, pp. 657–663.
- Couturier, O., E. M. Nguifo, et B. Noiret (2005). A hierarchical user-driven method for association rules mining. In *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI'05), Las Vegas, USA*.
- Couturier, O., J. Rouillard, et V. Chevrin (2007). An interactive approach to display large sets of association rules. In *Proceedings of the 2007 conference on Human interface, Berlin, Heidelberg*, pp. 258–267. Springer-Verlag.
- de Oliveira, M. et H. Levkowitz (2003). From visual data exploration to visual data mining : A survey. *IEEE Transactions on Visualization And Computer Graphics 9*(3), 378–394.
- Fayyad, U., G. Grinstein, et A. Wierse (2001). *Information visualization in data mining and knowledge discovery*. San Francisco : Morgan Kaufmann.

- Garey, R. et D. Johnson (1979). *Computers and Intractability, A Guide to the Theory of NP-completeness*. New York : Freeman.
- Hascoët, M. et M. Beaudouin-Lafon (2001). Visualisation interactive d'information. *Revue I3* 1(1), 77–108.
- Khchérif, R., M. M. Gammoudi, et A. Jaoua (2000). Using difunctional relations in information organization. *Information Sciences* 125, 153–166.
- Kuntz, P., R. Lehn, F. Guillet, et B. Pinaud (2006). Découverte interactive de règles d'association via une interface visuelle. In P. Kuntz et F. Poulet (Eds.), *Visualisation en extraction des connaissances*, Volume RNTI-E-7 of *Revue des Nouvelles Technologies de l'Information (RNTI)*. Cépaduès.
- Liu, Y. et G. Salvendy (2006). Design and evaluation of visualization support to facilitate association rules modeling. *International Journal of Human Computer Interaction* 21(1), 15–38.
- Mahanti, A. et R. Alhajj (2005). Visual interface for online watching of frequent itemset generation in APRIORI and ECLAT. *Fourth International Conference on Machine Learning and Applications, ICMLA 2005* 0, 404–410.
- Omicinski, E. R. (2003). Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering* 15(1), 57–69.
- Riguet, J. (1944). Relations binaires, fermetures et correspondances de Galois. *Bulletin de la société mathématique de France* 76, 114–155.
- Riguet, J. (1998). Deductibility and exactness. *Logic Journal of the IGPL* 6(2), 341–347.
- Robertson, G., M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, et M. V. Dantzich (1998). Data mountain : using spatial memory for document management. In *UIST '98 : Proceedings of the 11th annual ACM symposium on User interface software and technology*, New York, NY, USA, pp. 153–162. ACM.
- Wolpert, D. H. et W. Macready (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1), 67–82.
- Yang, L. (2003). Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates. In *Proceedings of ICCSA (1)*, pp. 21–30.
- Zaki, M. et B. Phoophakdee (2003). MIRAGE : A framework for mining, exploring and visualizing minimal association rules. Technical report, Rensselaer Polytechnic Institute, Computer Sciences Department, USA.

Summary

Visual mining as an emerging field is gradually witnessing approaches where visual capabilities are of use to mine manageable and useful knowledge. In this respect, the implication of the user at the core of the mining process seems to be one of the key factors in the success of any visual mining system. In this paper, we introduce a new anthropocentric visual mining approach for the extraction and the visualization of a reduced cover of association rules. Given that getting out such a reduced cover is NP-hard problem, we introduce a greedy algorithm that relies on correlation assessment metrics to flag discovered formal concepts as optimal. In addition, we present some snapshots illustrating the key features of the implemented visualization tool that relies on the virtual reality paradigm.

Exploration visuelle de la classification des données mixtes séquentielles

Rakia Jaziri^{1,2}, Mustapha Lebbah², Younès Bennani² and Jean-Hugues Chenot¹

¹ : Institut National de l'Audiovisuel
4, av. de l'Europe 94 366 Bry-sur-Marne
{rjaziri,Jhchenot}@ina.fr

² : Laboratoire d'Informatique de Paris-Nord, Université Paris 13
99, av. J-B Clément, F-93430 Villetaneuse
{younes.bennani, mustapha.lebbah, rakia.jaziri}@lipn.univ-paris13.fr

Résumé. L'accumulation de grands volumes de flux de données nécessite la mise en œuvre d'outils d'explorations telles que les méthodes de classification et de visualisation afin de comprendre les mécanismes impliqués dans les processus générant ces données. L'objectif de notre travail est de fournir un procédé d'identification des programmes télévisés et de détection des inter-programmes (plages de publicité, génériques, jingles...) à partir de données de répétition extraites de plusieurs flux télévisuels. Nous décrivons tout d'abord notre modèle de représentation du flux télévisuel. En particulier, nous insistons sur les outils permettant d'extraire un ensemble important d'indicateurs. Puis, nous montrons que l'utilisation des techniques de classification non supervisée permet de découvrir une segmentation pertinente qui devrait permettre à terme de délimiter automatiquement les programmes et de détecter les inter-programmes et leur enchaînement.

1 Introduction

La classification et le typage de flux importants de télévision posent de nombreux problèmes pratiques rencontrés par les centres d'archivage tel l'INA (Institut National de l'Audiovisuel), ou les organismes de contrôle de la diffusion tel le CSA (Conseil Supérieur de l'Audiovisuel). Dans le cas des centres d'archivage, l'INA a développé à cet effet des solutions techniques destinées à détecter des contenus audiovisuels, qui permettent de surveiller en temps réel 10 chaînes de télévision, et d'y identifier des segments de programmes d'une durée de quelques secondes similaires à des extraits d'une base de références de plusieurs milliers d'heures.

Les résultats de détection de répétitions sont très difficilement interprétables et la prise en compte des aspects temporels (généalogie des contenus) et structurels (motifs de répétitions) n'est pas évidente. Les solutions existantes sont avant tout manuelles ou semi-automatiques (Bennett 1999). De plus, la multiplication des chaînes, l'augmentation des quantités de données et l'apparition des appareils grand public d'enregistrement de télévision numérique poussent aussi à trouver des solutions de structuration automatique des flux télévisés.

Le principal objectif de cet article est de s'appuyer sur la masse des données récoltées pour extraire le maximum d'informations possible et d'ajuster les outils d'apprentissage pour que l'extraction des motifs puisse renvoyer à une pertinence significative.

A notre connaissance, il existe peu de travaux cherchant à classifier et étiqueter les programmes dans un flux télévisé d'une manière automatique et non supervisée. Une approche récente se base sur des requêtes pour classifier les données de répétitions et les affectant par la suite à des thèmes prédéfinis (Xiaomeng 2010). Certains segments ont été bien classés, mais cette approche a une erreur de classification majeure pour certaines classes telles que les programmes de télé-achat. Une approche plus sophistiquée consiste à utiliser des techniques de reconnaissance de parole pour identifier automatiquement des mots-clés ("word spotting") dans la bande-son du segment, ainsi reconnaître des programmes (Gelin 1996) et (Jones 1996). Bien que la reconnaissance de la parole présente encore des difficultés à résoudre (l'indépendance du locuteur, la taille du vocabulaire, la précision de la reconnaissance,...) des progrès spectaculaires ont été accomplis ces dernières années. On trouve cependant quelques études qui utilisent les sous-titres associés aux vidéos, mais ces approches sont souvent irréalisables, car les sous-titres sont rarement disponibles (Lin 2002) et (Brezeale 2006).

Par contre, de nombreux travaux se sont intéressés à la détection automatique des inter-programmes (tunnels de publicité, générique début ou fin, bande d'annonce), ce qui n'est pas exactement équivalent, car ces derniers ne forment qu'une partie spécifique de l'ensemble des programmes. Une approche générique et efficace est utilisée par (Duygulu 2004) qui détecte les publicités en tant que séquence répétitive. L'approche utilise une classification basée sur des attributs couleur et audio spécifiques au corpus utilisé. Une approche classique pour détecter les plages de publicités est la détection des images noires qui marquent la séparation entre deux publicités (Sadlier 2002). Cette détection est très simple à réaliser, mais produit de nombreuses fausses alarmes et est pour cette raison combinée avec d'autres attributs comme le silence (Sadlier 2002) et la fréquence du nombre de coupures. La séparation des publicités par une image noire n'est cependant pas une constante universelle. Certaines chaînes n'utilisent pas cette technique, et en France, les images de séparation sont blanches, bleues ou noires. Enfin, au-delà des difficultés propres à la classification, d'autres travaux s'intéressent à un niveau plus conceptuel du traitement, en particulier pour les applications de filtrage (Abiza 1996).

Cet article est organisé comme suit. La section 2 présente des méthodes de classification non supervisée pour la visualisation et l'étiquetage des flux de données. La section 3 décrit le dispositif expérimental et les évaluations. Enfin, la section 4 conclut et propose des perspectives d'évolution.

2 Classification et visualisation automatique des flux de données

Chercher à comprendre des données c'est chercher de l'information cachée dans un gros volume de mesures. La fouille de données permet d'appréhender le contenu de bases de données de taille importante. Elle constitue un ensemble de méthodes en vue de mettre en relief des relations existantes entre les objets et les paramètres Chavent (1998). Nous situons dans ce cadre la classification automatique non supervisée « clustering » qui regroupe l'ensemble des méthodes et algorithmes consistant à partitionner une population en un certain nombre de

groupes homogènes, appelés « clusters ». A chaque cluster est associé un représentant ou référent. Cette réduction du nombre d'observations à un ensemble de représentants pertinents est communément appelée quantification vectorielle.

Parmi les algorithmes les plus répandus, on citera celui des cartes auto-organisatrices de Kohonen (1997). Connu sous le nom de SOM (Self-Organizing Map). Ce modèle repose sur un algorithme non-supervisé compétitif, capable de projeter des données de grandes dimensions dans un espace à faible dimension (en général deux). Cette projection permet dans le cas où la carte est à une ou deux dimensions, une représentation visuelle et un jugement sur la qualité de la quantification. Les modèles des cartes auto-organisatrices sont souvent utilisés pour la visualisation et la classification.

Souvent la classification de Kohonen est suivie d'une classification automatique des référents de la carte de manière à les regrouper en cluster moins nombreux. Cette classification est souvent réalisée par des approches classiques telles que les approches hiérarchiques ou les approches par partitionnement type k-means (Chavent 1998, 1999).

A l'issue de l'étape de segmentation de la carte, il convient de déterminer quelle signification doit être accordée aux clusters résultants (film, pub, feuilleton...). Cette étape est validée par la caractérisation des clusters obtenus par l'expert du domaine. Cette tâche est beaucoup plus difficile en apprentissage non-supervisé vu l'absence des labels, l'importance du nombre de variables et la présence du bruit.

3 Expérimentations

Dans cette partie, nous allons discuter notre approche sur des bases de données réelles obtenues suite au prétraitement des données initiales de l'INA. La méthodologie utilisée se résume dans les différentes parties schématisées dans la figure 1. Dans la suite, nous détaillons ses différentes étapes.

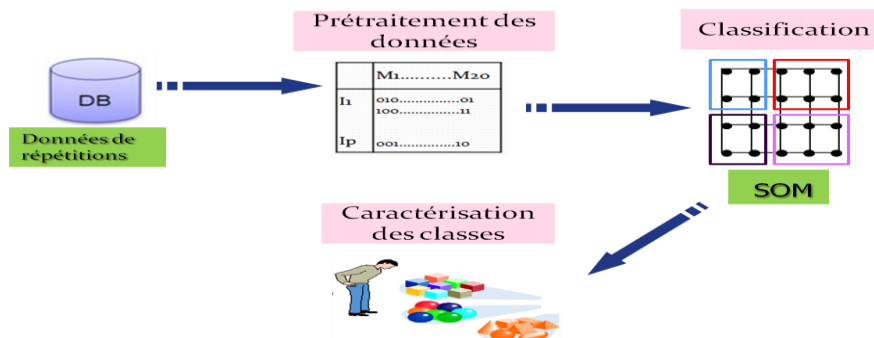


FIG. 1 – Organisation de l'architecture générale

Concernant l'évaluation, nous avons opté, dans un premier temps, pour une validation visuelle par un expert à partir d'un échantillon audiovisuel correspondant à chaque cluster. Dans un second temps et pour une meilleure évaluation des résultats obtenus, nous avons cherché à valider les résultats par l'approche de classification automatique, en considérant les informations obtenues par le guide de la télévision disponible sur ce lien :

<http://inatheque.ina.fr/Ina/ws/dltv/dlweb/general/MultiSearchNames9>. Une telle évaluation permettra de mesurer les pertinences des clusters obtenus, mais également leurs caractérisations.

Les jeux de données que nous avons utilisés n'incluent pas des informations sur les classes recherchées (étiquettes). L'objectif étant d'effectuer une classification non supervisée qui identifie correctement les structures cachées dans les données.

3.1 Prétraitement des données vidéo

Avant l'application des méthodes de classification automatique, un processus de prétraitement des données est appliqué. Cette étape permet d'enrichir les données de répétitions et d'extraire de l'information. En fait, l'enjeu n'est pas seulement la répétition, mais la manière dont les segments sont répétés. L'objectif est de pouvoir déterminer des types de répétitions qui possèdent une pertinence éditoriale pour la compréhension des flux vidéo et qui soient également extractibles par les moyens statistiques.

L'INA a exploité des contenus audiovisuels de dix chaînes de télévision afin de les transformer en une base de signatures de références et d'y identifier par la suite des segments de programmes d'une durée de quelques secondes similaires à des extraits d'une base de références de plusieurs milliers d'heures.

PGM1	TC11	TC12	PGM2
FR5.2009_07_13	00:01:10:04	00:02:19:09	FR5.2009_07_13
FR5.2009_07_13	01:46:54:19	01:48:03:24	FR5.2009_07_13
FR5.2009_07_13	04:01:35:06	04:02:17:24	FR5.2009_07_13
FR5.2009_07_13	03:53:16:16	03:53:59:07	FR5.2009_07_13
FR5.2009_07_13	00:53:58:22	00:54:27:14	FR5.2009_07_13
FR5.2009_07_13	05:00:54:13	05:01:23:05	FR5.2009_07_13
FR5.2009_07_13	03:58:46:24	03:59:12:17	FR5.2009_07_13
FR5.2009_07_13	05:51:20:23	05:51:44:11	FR5.2009_07_13
FR5.2009_07_13	04:09:07:21	04:09:31:08	FR5.2009_07_13
FR5.2009_07_13	08:10:23:20	08:10:42:14	FR5.2009_07_13
FR5.2009_07_13	07:44:58:13	07:45:17:05	FR5.2009_07_13

FIG. 2 – Un échantillon de la base de données initiales

Les résultats de cette comparaison se manifestent sous forme d'une table $T(n, 2)$ de n individus qui représentent la similarité entre les segments (FIG.2). Deux segments appartenant à une même ligne sont similaires (encadré en jaune) et chacun est identifié par la chaîne de diffusion (encadré en vert), la date (encadré en rouge), l'heure début (encadré en bleu) et l'heure de la fin (encadré en rose). Ces données sont obtenues après application des méthodes de comparaison de flux. Ils subissent une mise à jour fréquente et permanente.

Les données telles qu'elles se présentent sont redondantes, bruitées et mal organisées. Nous avons commencé par transformer la base en un ensemble de blocs constitué chacun par un label, un segment référent, une liste de p référents (p le nombre de segments qui sont similaires au segment référent) et des variables.

A l'issue de cette étape, nous avons développé un processus qui se base sur les données de répétitions organisées en blocs pour extraire de l'information concernant les segments. L'idée est de construire une base de données de dimension (n,m) où n représente le nombre de blocs et m l'ensemble des variables construites. Ainsi, chaque bloc i sera décrit par un représentant N_i qui est le segment le plus ancien (première apparition de la scène). Chaque référent est décrit avec 10 variables quantitatives et 10 variables qualitatives qui reflètent des informations pertinentes et utiles pour la classification. En effet, pour chaque séquence ou segment, on sélectionne un ensemble de variables en tenant compte des relations inter bloc (les séquences similaires) et intra bloc (les séquences précédentes, les séquences suivantes....), ci-dessous la description de certains types variables :

- La chaîne où il a été diffusé la première fois, et l'ensemble des différentes chaînes de diffusion, ce qui nous permet de distinguer les répétitions au sein d'un même canal et les répétitions entre plusieurs canaux.
- La durée de diffusion pour chaque segment (afin de pouvoir le classer parmi les segments de courte durée ou celle de la moyenne ou longue durée).
- Le nombre de répétitions dans chaque bloc.
- Les intervalles minimum, maximum, moyen et médian entre les segments d'un même bloc.
- L'heure pendant la journée : cette variable permet d'affecter le segment à une plage horaire (matin, matinée, midi, après-midi, soir et nuit).
- L'heure par rapport à une origine fixe pour vérifier s'il s'agit d'une répétition d'origine ancienne ou récente.
- Le jour de la semaine et le mois pour voir s'il existe des jours ou des mois spécifiques de diffusion (Jour Férié ou pas).
- La durée de la séquence précédente et suivante afin de détecter les couples génériques début et fin.

3.2 Classification et visualisation des données

Nous avons appliqué l'algorithme des cartes auto-organisatrices sur plusieurs bases d'apprentissages volumineuses, mais pour une meilleure visualisation nous avons choisi de présenter les résultats d'une base constituée de 3770 individus et 29 variables de dix chaînes télévisées pour la journée du 7 août 2010. La figure 3 montre la carte topologique de dimension 14×11 , où chaque chiffre indique le nombre d'exemples captés par chaque cellule. On constate que les exemples sont bien répartis sur la carte vu qu'elle se déforme pour couvrir tous les points de l'espace multidimensionnel pour représenter les données sur un espace à deux dimensions (FIG.5).

Afin d'analyser le comportement des variables sur les cellules, nous avons visualisé plusieurs cartes à la fois en indiquant la variation de la variable associée aux référents. En observant la figure 4 on peut constater les différentes corrélations apparentes entre les différentes variables. Par exemple, on constate clairement qu'il existe des corrélations entre la variable 1 et 2 ce qui est une évidence, car elles représentent la durée de répétition respecti-

vement en image et en seconde. On remarque aussi que lorsque la valeur de la variable 5 (intervalle minimum entre deux similarités) est très faible, la valeur de la variable 4 est très forte (Nombre de répétitions pendant la journée). Ce qui est prévisible, car lorsqu'une séquence se répète plusieurs fois dans la journée, on attend que l'intervalle minimum qui sépare ces répétitions soit faible. La même analyse peut être faite sur le reste des variables et nous permet de tirer des règles propres à chaque cluster. Par exemple, on peut dire que lorsque le nombre de répétitions est très élevé et l'intervalle maximum qui sépare ces répétitions est faible et les chaînes de diffusion sont multiples alors l'intervalle minimum et l'écart type sont faibles. Comme on peut constater aussi qu'il s'agit probablement d'une classe « publicité ».

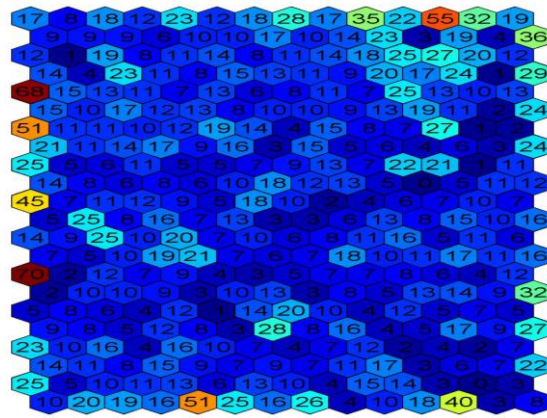


FIG. 3 – Carte SOM : cardinalité

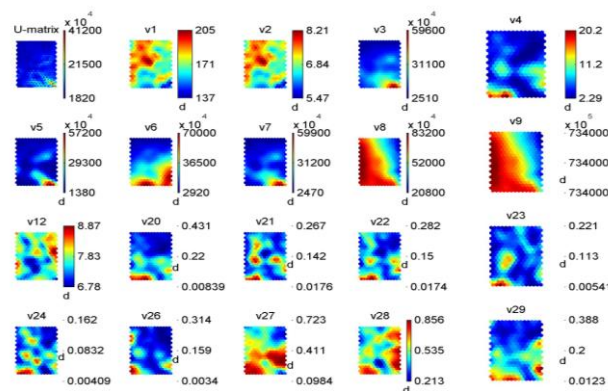


FIG. 4 – Le comportement des variables dans les clusters

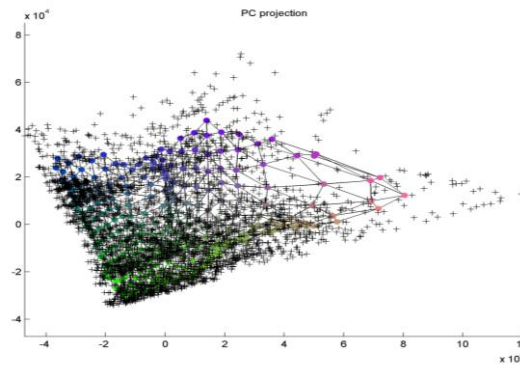


FIG. 5 – *Projection des données et de la carte topologique dans un espace 2D.*

Les cartes topologiques sont souvent combinées avec des méthodes de classification hiérarchique ou par partitionnement. Dans notre cas, la classification ascendante hiérarchique menée sur l'ensemble des vecteurs prototypes de la carte guidée par l'indice externe de Davies-Bouldin Davies (1979) conduit à retenir 14 classes. La figure 6 montre le découpage de la carte topologique en 14 clusters ; on constate clairement que chaque cluster est constitué de cellules voisines.

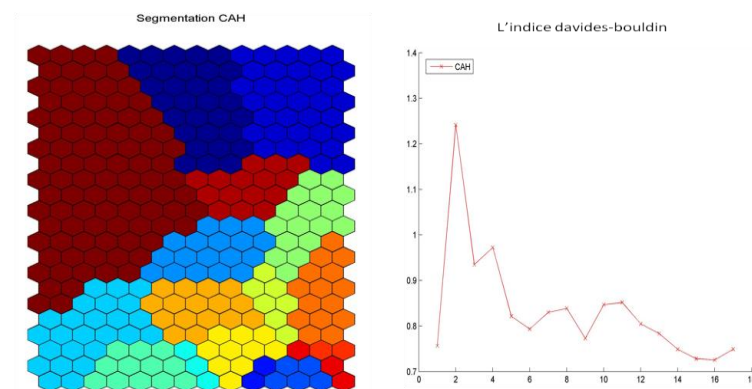


FIG. 6 – *Segmentation de la carte par CAH. La figure à droite représente la variation de l'indice de Davies-Bouldin pour différents découpage de la carte.*

Structuration automatique des flux télévisuels par apprentissage

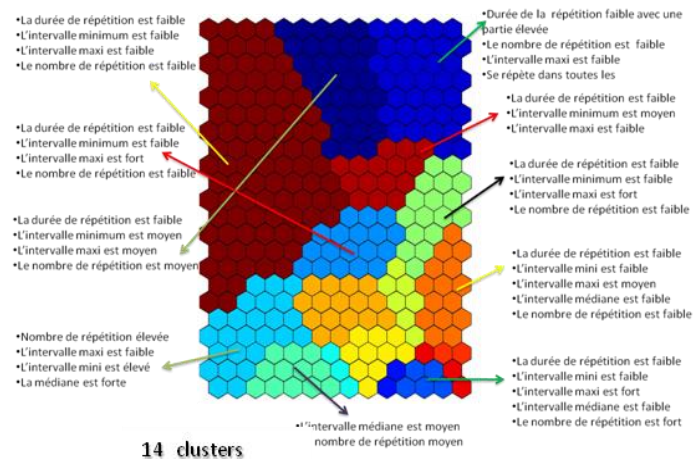


FIG. 7 – La caractérisation des clusters

La figure 7 montre les variables qui caractérisent le mieux chaque cluster. Afin de mieux caractériser les clusters, nous avons croisé les clusters avec le guide de la télé, en vérifiant à chaque fois le programme auquel appartient le segment. Nous affectons à chaque segment un ensemble d'images d'exemple. On constate clairement que les clusters sont constitués de données homogènes. La figure 8 montre pour chaque cluster un ensemble de prototypes d'image composant le cluster. On constate que les clusters sont constitués de contenus homogènes. Mais les inter-programmes (les jingles, les bandes d'annonces, les parrainages et les publicités) font parties d'un seul cluster, ce qui pourrait justifier une séparation plus fine.

1				Journal
2				Reportage
3				Générique
4				Tranche horaire
5				Desin animée
6				Publicités
7				Jeu
8				Téléfilm
9				Documentaire
10				Journal télévisé Brevé
11				Série Animation
12				Série
13				feuilleton
14				Vidéo Clip

FIG. 8 – Croisement des clusters avec les séquences correspondantes

Cette validation permet aussi d'illustrer le gain apporté par la classification par rapport au guide des programmes. Les manques de guide sont, non seulement les bornes des programmes, mais aussi la non-indication de nombreux programmes. Dans notre cas, la classification automatique se conforme parfaitement à la vérité terrain, alors que les indicateurs du guide sont assez décalés.

Les résultats sont très encourageants, produisant une bonne classification. Une difficulté importante est que la qualité des résultats dépend de la détection de répétitions, et du nombre de variables utilisées pour la classification. Il est nécessaire que la détection soit suffisamment complète pour que la classification soit effectivement de bonne qualité. Le nombre de variables utilisées va également être étendu.

4 Conclusion

L'objectif de notre travail est de montrer la faisabilité d'une solution de classification et d'étiquetage entièrement automatique de plusieurs flux de télévision. Des méthodes de classification automatique en mode non supervisé sont utilisées pour segmenter, puis étiqueter les flux grâce au guide de programme.

Bien qu'il reste encore beaucoup de travail à faire, les premiers résultats sont très encourageants. Différentes voies peuvent être envisagées en ce qui concerne l'amélioration de cette approche : exploitation de la détection des répétitions sur une durée plus longue (des feuilletons, des émissions complètes) et une structuration plus complexe (ensembles bande annonce, couple générique début - générique fin, rediffusion). Dans une phase ultérieure, nous envisageons de détecter des nouveautés (évolution de la typologie, apparition d'une nouvelle classe, événements exceptionnels).

Références

- Abiza.Y, Leger.A and Crehange.M, "Conceptual Modelling for Information Filtering in Broadcast Interactive Video Applications", Multimedia Modeling Conference, Toulouse, November 1996, pp 35-50.
- Darin Brezeale and Diane J. Cook, "Using Closed Captions and Visual Features to Classify Movies by Genre," Poster session of the Seventh International Workshop on Multimedia Data Mining (MDM/KDD2006), 2006.
- Davies.D. L and Bouldin.D.W, "A cluster separation measure", IEEE Transactions on Pattern Recognition and Machine Intelligence, pp224--227, VOL1, NO2, 1979.
- Sadlier.D., Sean Marlow, Noel OConnor, et Noel Murphy. Automatic tv advertisement detection from mpeg bitstream. 2002.
- Jones.G.J.F, Foote J.T, Sparck-Jones.K. and Young.S.J., "Retrieving spoken documents by combining multiple index sources", Proc. SIGIR'96, Zurich, August 1996, pp: 30-39.
- Vesanto.J and Alhoniemi.E Clustering of the self-organizing map, IEEE Transactions on Neural Network, VOL. 11, NO, pp. 586-600, 3, May 2000.

- Benabdeslem.K Approches connexionnistes pour la visualisation et la classification des séquences évolutives : Application aux données issues d'usages d'Internet. Université Paris 13, (Décembre 2003).
- Bennett .K. P, Fayyad .U, and Geiger .D, "Density-based indexing for approximate nearest-neighbor queries," in Proc. of Conf. on Knowledge Discovery in Data, 1999, pp. 233–243.
- Chavent.M, A monothetic clustering method .Pattern Recognition Letters 19, issue 11, 989-996. September 1998
- Chavent.M, Guinot .C, Lechevallier. Y, Tenenhaus.M, Méthodes divisives de classification et segmentation non supervisée : recherche d'une typologie de la peau humaine saine Revue de Statistiques Appliquées, XLVII (4), 87-99. 1999.
- Philippe Gelin and Christian Wellekens, "Keyword Spotting Enhancement for Video Soundtrack Indexing", Proc. ICSLP ,96, Philadelphia, PA, October 1996, vol 2, pp: 527–530.
- Pinar Duygulu, Ming yu Chen, et Alex Hauptmann. Comparison and combination of two novel commercial detection methods. Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME 2004), june 2004.
- Tufféry S. Data Mining et statistique décisionnelle L'intelligence dans les bases de données, Editions TECHNIP, Paris, 2005.
- Kohonen T. (1997), Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, 2nd edition.
- Wei-Hao Lin and Alexander G. Hauptmann, "A Wearable Digital Library of Personal Conversations", Joint Conference on Digital Libraries (JCDL), Portland, Oregon, USA, July 14-18, 2002
- Xiaomeng Wu, Ichiro Ide, Shin'ichi Satoh: PageRank with Text Similarity and Video Near-Duplicate Constraints for News Story Re-ranking. MMM 2010: 533-544

Summary

The accumulation of large volumes of TV flows requires the implementation of exploration tools such as clustering and visualization methods to understand the mechanisms involved in the processes generating the data. The objective of our work is to provide a method for identifying television programs and detecting cross-program links from repetition data extracted from several television streams. We first describe our representation model of the flow of television. In particular, we insist on tools to extract a large set of variables. Then, we show that the use of clustering techniques allow the unsupervised detection several classes of TV contents.

Visualisation des SVM et des changements de représentations induits par les noyaux

Matthieu Exbrayat*, Lionel Martin*

* Laboratoire d'Informatique Fondamentale d'Orléans
Université d'Orléans, BP 6759 F-45067 Orléans Cedex 2
Prenom.Nom@univ-orleans.fr
<http://www.univ-orleans.fr/lifo>

Résumé. Les noyaux permettent d'étendre l'espace des hypothèses de nombreuses méthodes en plongeant les données initiales dans un nouvel espace de représentation appelé espace de redescription. Cet espace n'est en général connu qu'au travers d'une fonction noyau qui permet de résoudre un problème sans avoir à calculer la représentation des objets dans ce nouvel espace. Dans cet article, nous nous intéressons à la visualisation des données dans ce nouvel espace, qui peut simplement être obtenue par une ACP à noyau, calculant une projection du nuage redécrit dans un espace observable. Cependant, cette méthode ne retient que les dimensions principales (en terme d'inertie du nuage), alors que le nombre de dimensions possibles est en général très grand (égal au nombre d'observations), ce qui donne une vision très partielle de ce nouvel espace.

Dans le cas de données supervisées, nous proposons d'étendre les axes de projection possibles, en considérant les dimensions discriminantes obtenues à l'aide de SVM. Dans un premier temps, nous proposons d'utiliser un tel axe qui sera ajouté à une représentation 2D obtenue par ACP. Nous obtenons ainsi une vue du nuage qui met en évidence la capacité du SVM à séparer les 2 groupes d'objets, offrant ainsi une vision partielle de la surface de séparation. Dans un deuxième temps, nous proposons de modifier la représentation des objets en ne gardant que les projections associées aux SVM, dans le même esprit que l'analyse discriminante. La visualisation 3D peut alors s'effectuer par une projection linéaire mettant en évidence la séparabilité des groupes dans l'espace de redescription.

1 Introduction

La visualisation constitue une étape importante dans l'exploration d'un jeu de données, du fait qu'elle fasse appel directement aux capacités d'interprétation visuelle du cerveau, et permette donc d'identifier rapidement et intuitivement des éléments de structuration du jeu étudié. Une stratégie simple de visualisation des données consiste à projeter celles-ci dans un espace de dimension observable (≤ 3). Si les données sont décrites par des attributs numériques, cette projection peut par exemple être obtenue par une analyse en composantes principales (ACP). Si on fournit en entrée une matrice de distances entre les objets, la projection peut être obtenue par positionnement multidimensionnel (MDS).

Toutefois, ces deux techniques, dans leur forme originelle, reposent sur un principe de projection linéaire. Elles consistent à fournir à l'utilisateur un point de vue "intéressant" sur l'espace d'origine, mais ne permet pas de réelle déformation de celui-ci. Des variantes non linéaires de ces deux approches ont donc été proposées, dont un grand nombre repose sur l'introduction de noyaux. Néanmoins, si les méthodes de projection à noyaux sont souvent au coeur des méthodes de visualisation, les représentations obtenues ne permettent pas d'apprécier les transformations induites par les noyaux utilisés.

Des améliorations visuelles peuvent être envisagées si l'on déplace légèrement le cadre de l'étude. Les méthodes de projection précédentes s'appliquent en effet dans un cadre non-supervisé. La plupart d'entre elles peuvent être adaptées pour tenir compte de classes et produire une représentation qui améliore la séparation des classes, selon des critères très variés. Parmi les très nombreuses approches existantes, celles reposant les séparateurs à vaste marge (SVM) sont particulièrement intéressantes. En effet, de par leur nature "géométrique", les SVM peuvent également faire l'objet d'une visualisation directe dans les cas les plus simples –objets décrits dans un espace à deux ou trois dimensions–, et indirecte la plupart du temps. La visualisation directe d'un SVM consiste à le représenter comme une droite (en 2D) ou un plan (en 3D). La visualisation indirecte s'appuie en général sur une coloration des objets –en 3D– ou de l'espace –en 2D.

Une visualisation totalement différente des données et du SVM associé peut toutefois être envisagée : plutôt que de chercher à représenter le SVM dans la projection 3D, celui-ci peut servir de base au calcul des dimensions de projection. Dans leurs travaux, Maszczyk et Duch (2008) proposent ainsi un nouveau modèle de projection : étant données 2 classes, un SVM linéaire est construit et les objets sont projetés sur un axe orthogonal à la surface de séparation. Les données peuvent être alors projetées dans un sous-espace orthogonal à cet axe et le processus peut être répété jusqu'à convergence d'un terme d'erreur. Cette méthode est adaptée aux SVM linéaires, mais les auteurs proposent une méthode pour approcher la projection orthogonale dans le cas de SVM non-linéaires.

Dans cet article, nous proposons une nouvelle méthode utilisant les SVMs dans le même esprit, et permettant de visualiser ceux-ci en 3D. Cette méthode consiste à modifier la représentation des données en y intégrant des caractéristiques induites par les SVM : à partir d'une projection linéaire des données (sans transformation de l'espace de représentation initial, autre qu'une perte d'information) en dimension 2, nous proposons d'ajouter une 3ème dimension calculée à l'aide de la projection (éventuellement non linéaire) sur l'axe orthogonal à la surface de séparation. Il est ainsi possible de visualiser partiellement la surface de séparation associée à un SVM et d'observer les transformations de l'espace induites par le noyau.

En étendant ce principe, nous proposons également de produire une nouvelle représentation construite à partir des caractéristiques induites par un ensemble de SVM, qu'ils soient kernelisés ou non. Enfin, ces fonctionnalités ont été intégrées au logiciel Explorer3D, qui offre en plus la possibilité de comparer plusieurs représentations de façon interactive, en affichant des informations complémentaires simultanément dans les différentes représentations, comme par exemple les labels des objets ou des structures caractéristiques (e.g. arbre minimum de recouvrement). Ces comparaisons sont possibles y compris pour les noyaux usuels (ISOMAP, LLE, ...).

Cet article s'articule comme suit : en section 2 nous revenons de manière plus formelle sur les notions de SVM, noyaux et méthodes de projection. En section 3, nous présentons

et comparons différentes méthodes et outils de projection. En section 4, nous nous penchons sur les éléments de visualisation d'un SVM. En section 5 nous étudions une visualisation par projection suivant les axes orthogonaux aux SVM. Enfin, nous concluons et envisageons diverses perspectives en section 6.

2 Préliminaires

2.1 Noyaux

Les noyaux sont aujourd'hui intégrés à de nombreux problèmes d'apprentissage (Schölkopf et Smola (2002)) pour lesquels on dispose d'une méthode de résolution où les données d'origine n'apparaissent que via leurs produits scalaires $\langle x_i, x_j \rangle$. Il est alors possible d'utiliser une transformation de l'espace d'origine afin de produire un espace de plus grande dimension, dans lequel on espère voir émerger des régularités non perceptibles dans l'espace d'origine.

Cette transformation de l'espace est usuellement représentée par une application ϕ qui permet de "plonger" les points de l'espace initial dans un nouvel espace de redescription. Ainsi, chaque observation x_i peut être remplacée par son image $\phi(x_i)$ dans le problème initial. $\phi(x_i)$ peut être considérée comme une représentation de x_i dans un espace de dimensionnalité plus élevée que l'espace d'origine. Chaque dimension de l'espace d'arrivée peut être définie en fonction des dimensions de l'espace d'origine. Néanmoins, les coordonnées des objets dans ce nouvel espace ne sont jamais calculées. En effet, l'intérêt de ce type de transformation réside dans la capacité à calculer les produits scalaires dans l'espace d'arrivée $\langle \phi(x_i), \phi(x_j) \rangle$ sans avoir à calculer explicitement les nouvelles représentations $\phi(x_i)$. S'il est possible de trouver une application "noyau" :

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

telle que $k(x_i, x_j)$ peut être calculé efficacement, alors il suffit de remplacer les produits scalaires $\langle x_i, x_j \rangle$ du problème d'origine par $k(x_i, x_j)$, pour pouvoir résoudre ce problème dans l'espace d'arrivée. On parle dans ce cas "d'astuce du noyau". De très nombreux noyaux de ce type ont été proposés comme le noyau polynomial, RBF, sigmoïde, ... (Schölkopf et Smola (2002)). Cependant il n'est pas toujours aisé d'obtenir une représentation des changements qu'ils induisent.

2.2 MDS & ACP à noyau

L'analyse en composante principale ou ACP (Hotelling (1933)) est une méthode de réduction de dimensions qui calcule la projection linéaire maximisant la variance du nuage projeté. Dans ce cas, on dispose en entrée d'une représentation des données dans un espace \mathbb{R}^d et cette méthode fournit une représentation dans un espace $\mathbb{R}^{d'}$ avec $d' < d$; cette méthode est souvent utilisée pour la visualisation. Elle peut facilement être utilisée avec un noyau (Schölkopf et al. (1997)). On obtient alors une projection non linéaire des données initiales, qui peut également être vue comme une projection linéaire des données après plongement dans le nouvel espace de représentation. Il existe une grande variété de noyaux usuels utilisables dans ce cadre, tels les noyaux polynomial, RBF ou sigmoïde. Il existe naturellement diverses propositions dans le prolongement de la KACP. Citons par exemple les travaux de Chen et al. (2008), qui étendent

le cadre de la KACP à l'analyse des corrélations canoniques (ACC) ainsi qu'au clustering à noyau.

Le positionnement multidimensionnel ou MDS (Cox et Cox (1994)) est une méthode également très utilisée pour la visualisation. Elle permet également d'obtenir une représentation d'un ensemble d'observations dans un espace $\mathbb{R}^{d'}$, telle que dans cet espace, les distances euclidiennes soient le plus proches possible des distances fournies en entrée. On dispose donc en entrée d'une matrice de distances entre toutes les paires d'observations. Si on suppose que les distances fournies sont des distances euclidiennes (dans un espace inconnu), la résolution de ce problème est très analogue à celui de l'ACP ; on parle alors de MDS linéaire.

Il existe de nombreuses extensions non linéaires à ce problème, reposant par exemple sur les méthodes ISOMAP, LLE ou Eigenmaps. Notons que ces transformations non linéaires sont en général motivées et guidées par la recherche d'une organisation particulière des données. Ainsi, la mise en oeuvre d'ISOMAP (Tenenbaum et al. (2000)) implique qu'une structuration des objets le long d'une surface complexe (variété) est soupçonnée, et que l'on tente ainsi de restituer au mieux la distance géodésique (et non euclidienne) entre les objets et leurs plus proches voisins. La projection résultante se fera alors sur une variété de plus faible dimension. Le plongement localement linéaire ou LLE (Roweis et Saul (2000)) essaie également de restituer dans l'espace projeté, des propriétés locales, puisque chaque observation sera "reconstruite" à partir de ses plus proches voisins dans l'espace d'origine.

Les deux cadres de l'ACP et du MDS sont très voisins, la plupart des méthodes de MDS pouvant être vues comme des KACP dotées d'un noyau approprié (Bengio et al. (2003)). Ainsi, dans la suite de cet article, l'"ACP à noyau" désignera aussi bien l'ACP que le MDS (linéaire ou non). Cependant, il existe une importante différence d'objectif entre ces deux familles de transformations : dans le cas des MDS non-linéaires, le noyau associé a été conçu pour capter une transformation géométrique qui peut être interprétée. Pour les autres noyaux (polynomial, RBF, ...), leur utilisation est plus souvent motivée par leur capacité à étendre l'espace de recherche, ce qui peut par exemple aboutir à l'amélioration des performances d'un classifieur ; en revanche, leur interprétation géométrique est moins aisée.

2.3 SVM

Les séparateurs à vaste marge ou SVM (Vapnik (1995)) permettent de classer des objets dans un contexte supervisé. Ici encore, les observations sont décrites dans un espace \mathbb{R}^d avec en plus, un identifiant de classe. L'objectif consiste alors à trouver l'équation $h(x) = 0$ d'une surface de séparation qui soit équitablement éloignée des populations de ces deux classes. On peut identifier deux catégories de SVM, suivant le type de surface recherchée : les SVM "linéaires" et "non linéaires".

Dans le cas d'un SVM "linéaire", on cherche une surface séparatrice dans l'espace de description initial des objets –que l'on suppose décrits par des attributs numériques. Cette surface est alors un hyperplan. La classification d'une observation –ou objet– supplémentaire x_i est alors réalisée en observant de quel côté de la surface il se situe, c'est à dire en considérant une projection linéaire de x_i sur un axe orthogonal à cet hyperplan et en notant le signe de cette projection. En d'autres termes, pour toute observation x_i , $h(x_i)$ peut être calculé et son signe permet de classer x_i . Notons que cette notion de projection permet d'utiliser les SVM comme une méthode de calcul d'un sous-espace de projection dans un contexte supervisé, comme

l'ACP en est une dans le cas non-supervisé. Un tel SVM est efficace si les deux populations sont effectivement séparables par un hyperplan. Dans la cas d'une organisation plus complexe dans l'espace d'origine, des approches plus sophistiquées, dites "non linéaires", doivent être envisagées.

Les SVM "non-linéaires" s'appuient sur la recherche d'une surface séparatrice plus complexe. Une façon naturelle d'envisager de tels SVM consiste à effectuer une transformation non linéaire de l'espace de représentation en ayant recours à des méthodes à noyau. En réalité, on peut considérer qu'il s'agit toujours de rechercher un hyperplan, mais à partir d'un nouvel espace de représentation (différent de celui utilisé pour la description des données). En général, ce nouvel espace de représentation n'est pas explicite, mais est spécifié au travers d'une fonction noyau qui permet de calculer un produit scalaire dans cet espace. Il devient alors difficile d'avoir une vision intuitive de la transformation de l'espace associée à un noyau.

Concernant la visualisation de SVM, il existe principalement 2 approches. La première consiste non pas à visualiser la surface de séparation mais à représenter des indicateurs sur une classification d'un objet en particulier. Parmi ces méthodes, on peut citer les "nomograms" (Jakulin et al. (2005)) qui permettent des visualisations comme celles de la figure 1.

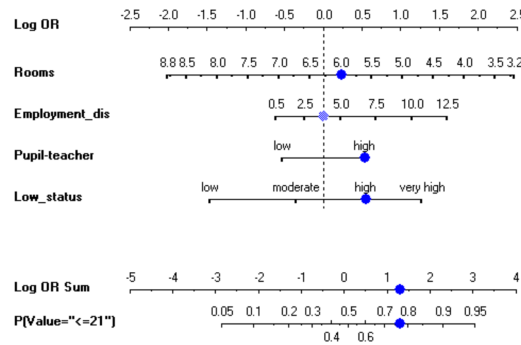


FIG. 1 – Exemple de nomogramme associé à un SVM (Jakulin et al. (2005)).

La seconde famille de méthodes de visualisation a pour objectif de représenter la position relative des objets par rapport à la surface de séparation. Usuellement, il existe 2 façons de représenter visuellement un SVM dans une scène 2D ou 3D : si les données sont en 3D et que le SVM est linéaire, alors le SVM peut être matérialisé par un plan (cf Fig. 2 à droite) ; dans le cas de données en 2D, il sera matérialisé par une droite. Il est également possible, pour des données en 2D, de choisir une couleur pour chaque demi-espace (séparés par le SVM) : chaque point peut alors être représenté par sa couleur (selon le côté où il se trouve), l'intensité de celle-ci correspondant à sa distance à la surface de séparation. Ce modèle permet de colorer chaque point (pixel) représenté, donnant ainsi une vision précise de la surface de séparation (cf Fig. 2 à gauche). Cette technique est cependant restreinte aux seules représentations 2D, car chaque pixel est coloré. Dans le cas d'une vue 3D, cette coloration globale rend la vue peu lisible (même si l'on peut envisager de jouer sur la transparence des voxels). Il convient alors de limiter la coloration aux seuls objets de l'étude ; on perd alors la possibilité d'observer clairement la surface de séparation. Une approche hybride peut toutefois être utilisée

Visualisation 3D de SVM et noyaux

dans le cas où les données sont initialement dans un espace 3D : il est possible de déterminer un ensemble de points appartenant à la surface de séparation et de représenter cette surface. Cet ensemble de points est déterminé par un parcours exhaustif de l'espace 3D discrétisé, ce qui limite cette approche à 3 dimensions mais fournit des visualisations très intéressantes des surfaces de séparation. Cette méthode est illustrée par la figure 3.

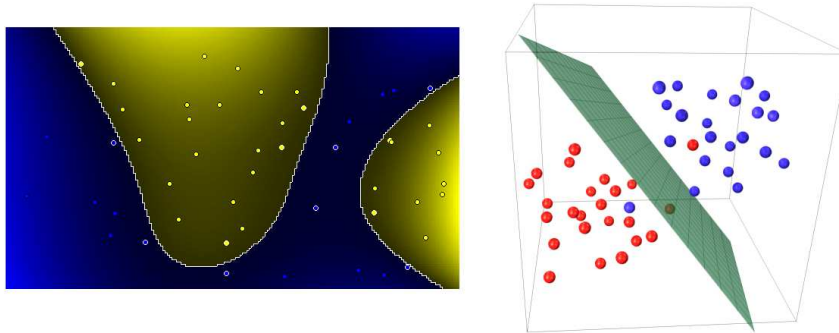


FIG. 2 – Exemples de visualisation de SVM : à gauche SVM non-linéaire en 2D (<http://svm.dcs.rhbc.ac.uk/pagesnew/GPat.shtml>), à droite SVM linéaire en 3D (<http://www.bwaldvogel.de>).

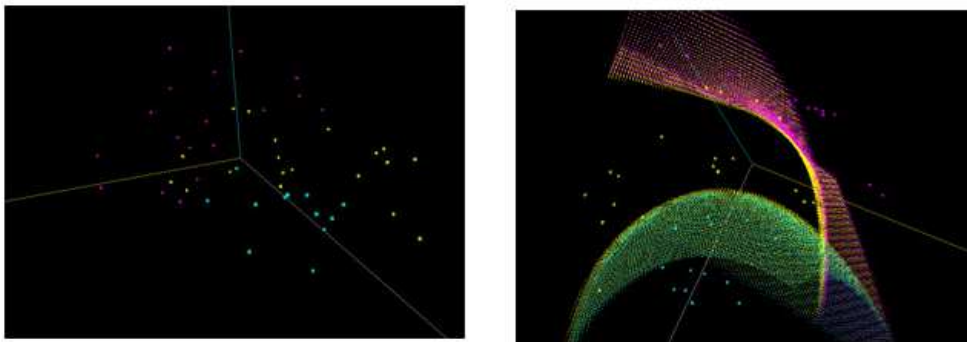


FIG. 3 – Exemples de visualisation de la surface de séparation pour des données en 3D (<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/svmtoy3d/>).

3 Comparaisons de représentations

La façon la plus directe de visualiser les changements induits par un noyau consiste à utiliser une ACP à noyau et à en visualiser le résultat. Cette opération est possible dans notre

outil Explorer3D qui propose une visualisation en 3 dimensions, qui permet en plus de comparer avec la représentation sans noyau (ACP standard). En juxtaposant ces 2 vues, il est possible de faire apparaître simultanément des informations complémentaires comme des labels, des couleurs associées à des attributs de classe, des groupes, ... Dans l'exemple de la figure 4, le jeu de données initial est uniformément distribué dans une sphère en dimension 4 (à gauche). La couleur représente la valeur d'un attribut complémentaire (ici la distance au centre). On observe que le noyau polynomial de degré 2 projette prioritairement dans un cône (à droite) et que l'axe principal est clairement associé à la distance au centre.

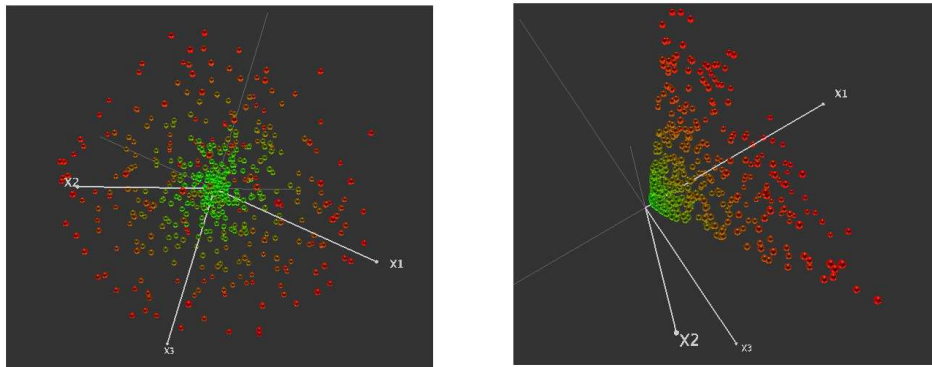


FIG. 4 – Comparaison de visualisations d'un même jeu de données : à gauche ACP standard, à droite ACP avec un noyau polynomial de degré 2.

Dans cet outil, il est également possible d'afficher des informations de structure comme par exemple l'arbre minimum de recouvrement (qui est associé à un clustering hiérarchique) ou encore le graphe de voisinage (souvent utilisé dans les MDS non-linéaires). Dans la figure 5, les données sont distribuées sur une hélice (à gauche), cette distribution étant approximative (un faible bruit a été ajouté). Dans la figure de droite, on peut observer la représentation obtenue avec le noyau ISOMAP (pour un voisinage de taille 4). Pour une hélice sans bruit, avec ce type de noyau, nous devrions obtenir un alignement parfait des points. Le bruit induit ici quelques anomalies dans le graphe de voisinage, la représentation obtenue faisant apparaître ces anomalies. Dans cette figure, les arêtes représentent le graphe de voisinage pour 4 voisins (pour chaque objet, une arête le relie à ses 4 plus proches voisins).

Ces comparaisons constituent un outil intéressant pour observer les principales transformations induites par le noyau sur un jeu de données. Cependant, ces transformations sont très dépendantes de la distribution des données, il n'est donc pas possible de généraliser ces observations. Par ailleurs, cette visualisation se limite aux 3 principales dimensions de l'analyse, alors que dans le cas des noyaux usuels celle-ci peut en retourner autant que d'objets.

Dans la section suivante, nous proposons d'observer d'autres dimensions de projection, qui ne sont pas nécessairement pertinentes en terme de variance expliquée, mais qui représentent, dans le cas supervisé, un axe de discrimination entre 2 classes.

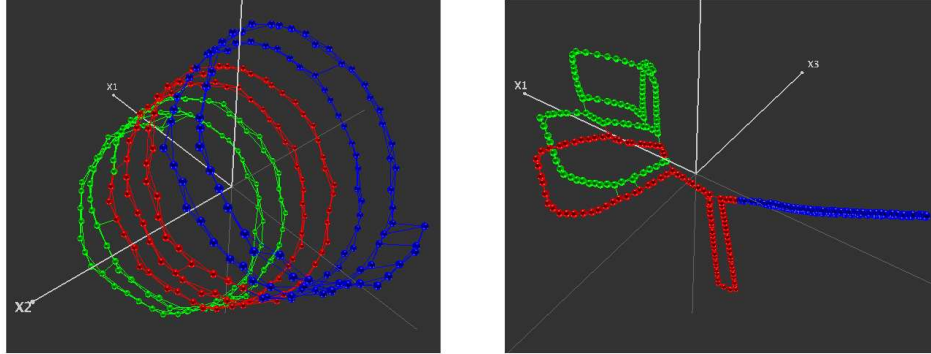


FIG. 5 – Comparaison de visualisations d’un même jeu de données avec le graphe de voisinage : à gauche ACP standard, à droite ACP avec un noyau ISOMAP (4 voisins).

4 Visualisation et SVM

Dans le cas supervisé, un SVM permet d’obtenir une “surface de séparation” entre 2 groupes d’objets, ayant pour équation $h(x) = 0$. S’il n’est pas toujours possible d’obtenir explicitement cette équation, on peut en revanche calculer $h(x_i)$ pour n’importe quelle observation x_i . Ainsi la valeur $h(x_i)$ peut être vue comme la projection de l’observation x_i sur un axe orthogonal à la surface de séparation. Cette projection est par exemple utilisée dans Maszczyk et Duch (2008).

Nous proposons ici de construire une représentation composée de :

- 2 dimensions issue d’une ACP (les 2 dimensions principales),
- 1 dimension issue d’un SVM. Pour cela, nous construirons soit un SVM “1 contre 1” qui construit un séparateur entre 2 classes choisies par l’utilisateur, soit un SVM “1 contre tous” qui construit un séparateur entre les objets d’une classe et tous les autres objets.

Nous obtenons ainsi une visualisation 3D qui permet à la fois une observation du nuage dans l’espace de redescription, et en même temps qui permet d’observer la répartition réalisée par le SVM, montrant les erreurs ainsi qu’une vision partielle de la surface de séparation.

Dans l’exemple de la figure 6, les données sont réparties en 3 classes correspondant à des sphères creuses concentriques (à gauche). En utilisant un SVM polynomial de degré 2, on obtient une séparation parfaite entre la classe la plus au centre et les autres objets.

En revanche, cette représentation permet d’observer des situations de sur-apprentissage, qui peuvent se produire en utilisant certains noyaux. Dans l’exemple de la figure 7, les points ont été générés aléatoirement dans une sphère, avec une distribution uniforme. Les labels de classe ont été affectés aléatoirement. En utilisant un noyau RBF on obtient un séparateur parfait.

La figure 8 présente le jeu de donnée “glass” de l’UCI repository. L’axe vertical correspond à la projection associée au SVM (avec un noyau RBF) discriminant la classe 2 (en rouge) de toutes les autres. Cette figure montre que ces classes ne sont pas parfaitement séparées, on peut observer en particulier une zone comportant des erreurs de classification (au centre légèrement à droite, difficile à observer dans une vue 2D !).

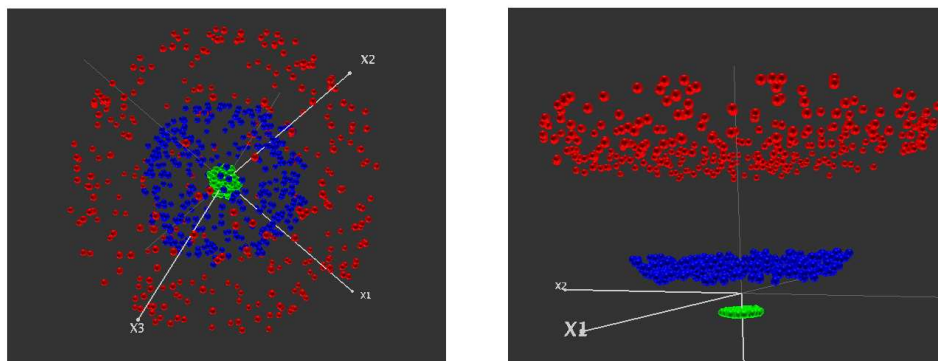


FIG. 6 – Données initiales (à gauche); à droite, la 3ème dimension est associée à un SVM avec noyau polynomial entre la classe la plus au centre et le reste.

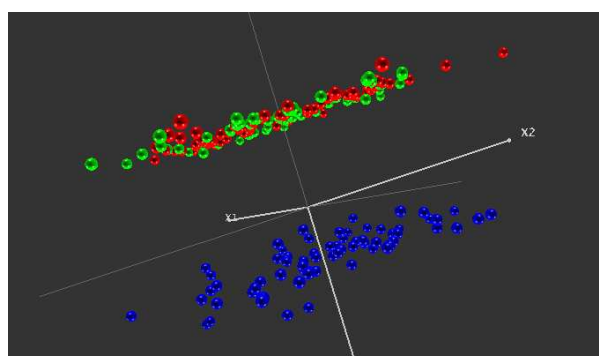


FIG. 7 – SVM avec noyau RBF séparant parfaitement des objets générés aléatoirement.

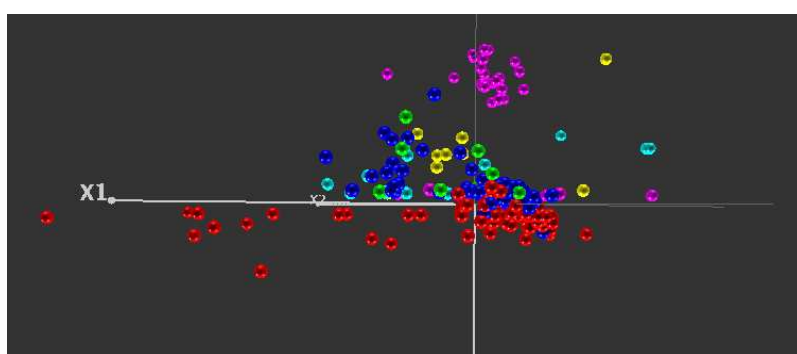


FIG. 8 – Projection du jeu de données “glass” (noyau RBF).

5 Généralisation des changements de représentation

Nous proposons maintenant de généraliser l'utilisation de la projection associée à un SVM pour calculer une nouvelle représentation. L'idée consiste à construire des SVM "1 contre tous" (resp. "1 contre un"), pour chaque classe (resp. couple de classes). Étant données k classes, nous obtenons ainsi une représentation dans un espace de dimension k (resp. $k(k-1)/2$). Cette représentation correspond aux axes de projection non-linéaires les plus discriminants, obtenus à l'aide de SVM. On peut considérer que dans ce nouvel espace, les groupes seront assez bien séparés.

Comme cet espace peut être de dimension supérieure à 3, nous proposons de le visualiser en utilisant une ACP (sans noyau). La résolution est différente de l'analyse discriminante mais l'objectif global est assez voisin : il consiste à chercher une représentation qui sépare les classes, l'analyse discriminante tentant également de regrouper les objets d'une même classe. Cependant, nous bénéficions dans notre cas du pouvoir de séparation de projections non-linéaires.

Notons que dans la résolution de l'ACP, c'est la somme des distances à l'origine qui est maximisée, ce qui est équivalent, après centrage, à la somme des distances à la moyenne. Dans notre cas, il est préférable de ne pas centrer les données, ainsi l'origine correspond à l'intersection des axes de séparation maximale, ce qui conduit à chercher une représentation qui restitue au mieux les séparations induites par les SVM.

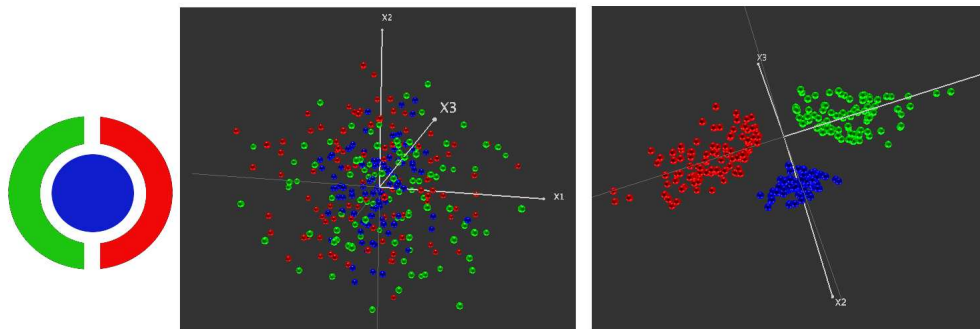


FIG. 9 – Projection à l'aide de SVM "1 contre tous" d'un jeu de données artificiel. à gauche : vue schématique de l'organisation des 3 classes ; au centre : projection par ACP ; à droite : projection suivant les axes des trois SVM "1 contre tous".

Nous utilisons de nouveau un exemple jouet pour illustrer cette construction. Dans la figure 9, la partie gauche schématise la répartition des 3 classes : les objets d'une des classes forment une hyperboule, et sont entourés par ceux des deux autres classes, structurées chacune sous forme d'une demi-hyperboule creuse. Nous avons généralisé cette construction dans un espace à 6 dimensions, auquel ont été ajoutées 3 dimensions de bruit. Au centre de la figure est présenté le jeu de données à l'aide d'une ACP linéaire. Si nous utilisions des noyaux polynomi- aux (de degré 2), chaque SVM obtenu dans un mode "1 contre tous" séparerait parfaitement les objets (une classe contre les 2 autres). Cependant, pour chacune des 3 représentations obtenues,

nous obtiendrions une classe isolée et les 2 autres mélangées. En utilisant la méthode de projection que nous proposons, nous obtenons une représentation où les 3 classes sont parfaitement séparées.

6 Conclusion et perspectives

Nous avons proposé dans cet article un moyen de visualiser des données dans l'espace de redescription associé à un noyau. Pour cela nous avons utilisé des méthodes de projection à noyaux comme l'ACP ou les SVM. Notons qu'il est possible d'utiliser d'autres méthodes de projection, à condition qu'elles puissent être utilisées avec un noyau.

Les expériences permettent d'observer des organisations de l'espace de redescription, cependant il faut garder à l'esprit que dans notre contexte, nous n'obtenons qu'une vision très partielle des données puisque dans ce nouvel espace, le nombre de dimensions "observables" peut être grand (de l'ordre du nombre d'observations). Elles permettent cependant de visualiser la séparation induite par le SVM, et peuvent aussi mettre en évidence les limites de l'utilisation des noyaux comme par exemple des cas de sur-apprentissage.

Pour poursuivre ce travail, nous proposons d'utiliser la nouvelle description des données, décrite dans la section 5, pour réaliser une classification. En effet, cette nouvelle description peut induire une décision différente de celle produite par une approche du type multiclass-SVM ; nous envisageons donc de comparer les performances de ces classifieurs.

Références

- Bengio, Y., J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, et M. Ouimet (2003). Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *NIPS*.
- Chen, C.-h., W. Härdle, A. Unwin, Y.-c. I. Chang, Y.-J. Lee, H.-K. Pao, M.-H. Lee, et S.-Y. Huang (2008). Data visualization via kernel machines. In *Handbook of Data Visualization*, Springer Handbooks Comp.Statistics, pp. 539–559. Springer Berlin Heidelberg.
- Cox, T. et M. Cox (1994). *Multidimensional scaling*. London : Chapman and Hall.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24, 417–441.
- Jakulin, A., M. Mozer, J. Demšar, I. Bratko, et B. Zupan (2005). Nomograms for visualizing support vector machines. In R. Grossman, R. J. Bayardo, et K. P. Bennett (Eds.), *KDD*, pp. 108–117. ACM.
- Maszczyk, T. et W. Duch (2008). Support vector machines for visualization and dimensionality reduction. In V. Kurková, R. Neruda, et J. Koutník (Eds.), *ICANN (1)*, Volume 5163 of *Lecture Notes in Computer Science*, pp. 346–356. Springer.
- Roweis, S. et L. Saul (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500), 2323–2326.
- Schölkopf, B. et A. J. Smola (2002). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press.

- Schölkopf, B., A. J. Smola, et K.-R. Müller (1997). Kernel principal component analysis. In *ICANN*, pp. 583–588.
- Tenenbaum, J. B., V. Silva, et J. C. Langford (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290(5500), 2319–2323.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York : Springer.

Summary

Kernels are a mean to extend the description space of objects by computing a new set of features, called re-description space. This latter is usually manipulated through a kernel function, the projected coordinates of objects being thus seldom computed. In this paper we focus on the visualization of objects in this new space, computing a low-dimension projection of this re-description space by the mean of a kernel PCA. Nevertheless, such a strategy does only highlight the principal dimensions, while the re-description space is usually a high-dimensional one. Visualization does consequently limit to a very small part of the re-description space.

Considering supervised data, we propose to explore additional projection dimensions, relying on the discriminative dimensions related to a SVM. We first add such an axis as a third dimension to a 2D PCA projection. We thus have a view on the SVM ability to discriminate together with a partial view of the separation surface. Second, we focus more specifically on the only projections along several axis induced by the SVM, in a manner that is somehow similar to discriminant analysis. 3D visualization can thus correspond to a linear projection that highlights how much the re-description space allows for an effective discrimination of classes.

Tulip: a Scalable and Adaptive Graph Visualization Framework

David Auber, Romain Bourqui, Maylis Delest, Jonathan Dubois, Ludwig Fiolka, Charles Huet, Antoine Lambert, Patrick Mary, Morgan Mathiaut, Guy Melançon, Bruno Pinaud ^{*,**}

^{*}CNRS UMR 5800 LaBRI, Campus Université Bordeaux I
@labri.fr

<http://www.labri.fr>

^{**}INRIA Bordeaux Sud-Ouest, Campus Université Bordeaux I
@inria.fr

<http://www.inria.fr/bordeaux>

Résumé. The Graph Visualization Framework Tulip now enjoys 10 years of user experience, and has matured its architecture and development cycle. Originally designed to interactively navigate large graphs, the framework integrates state-of-the-art software engineering concepts and good practices. It offers a large panel of graphical representations (traditional graph drawing as well as alternate representations). Tulip is most useful in a data mining and knowledge discovery context, allowing users to easily add their own data analysis and computing routines through its plug-in architecture for C++ and Python. The most recent version of Tulip eases its update and adjunction of new functionalities as it dynamically connects to plug-in servers each time it is run.

1 Introduction

Most research efforts in data mining and knowledge extraction and representation require experimentation and validation. To this end, our group develops the Tulip Graph Visualization Framework originally designed and developed by David Auber (Auber, 2003). Tulip offers a C++ plug-in mechanism easing the development and adjunction of new algorithms (computing graph statistics, graph drawing or graph clustering). This actually is a main feature of the Tulip platform (Auber et Mary, 2006).

Tulip implements astute data structures, data management and filtering/inheritance mechanisms. Its rendering engine relies on OpenGL, while its GUI rests on Nokia's QT library. The main interaction paradigm offered by Tulip is the computation and direct manipulation of graph hierarchies, making it unique among all available graph visualization platforms such as Pajek (Batagelj et Mrvar, 2002; de Nooy et al., 2004), GraphViz (Gansner et North, 2000; Ellson et al., 2002), Dynagraph (Ellson et al., 2003), the InfoVis Toolkit (Fekete, 2004), pre-fuse (Heer et al., 2005), and others ¹.

1. See <http://gvsr.polytech.univ-nantes.fr>

Tulip: a Scalable and Adaptive Graph Visualization Framework

Tulip's architecture eases the construction of stand-alone and specialized applications. An application targeted at domain specific data can be realized through the selection of building blocks (data structures, existing plug-ins), customization/development of a QT-based GUI, and adjunction of specific plug-ins all integrated into a *Tulip perspective* (in this respect, Tulip's mimics the architecture underlying the Eclipse IDE developed by the Eclipse Foundation²).

2 How to use Tulip ?

Fig. 1 illustrates a typical scenario when performing data exploration and analysis using Tulip. Nodes represent poker players and there is an edge between two nodes a and b if a gives money to b . While the left pane gives access to several graph properties and tuning parameters, the right panes shows several views on the data : standard node-link graph drawing (top left) with an interactive magnifying glass ; standard tabular sheet (bottom right) ; parallel coordinates (here shown circularly) ; self-organizing maps (bottom left) and histograms, scatterplot matrix (top right). Tulip is incredibly efficient at synchronizing views and keeping properties coherent between views. For instance, applying colormaps through histograms or selecting nodes through self-organizing maps immediately transfers to all other views.

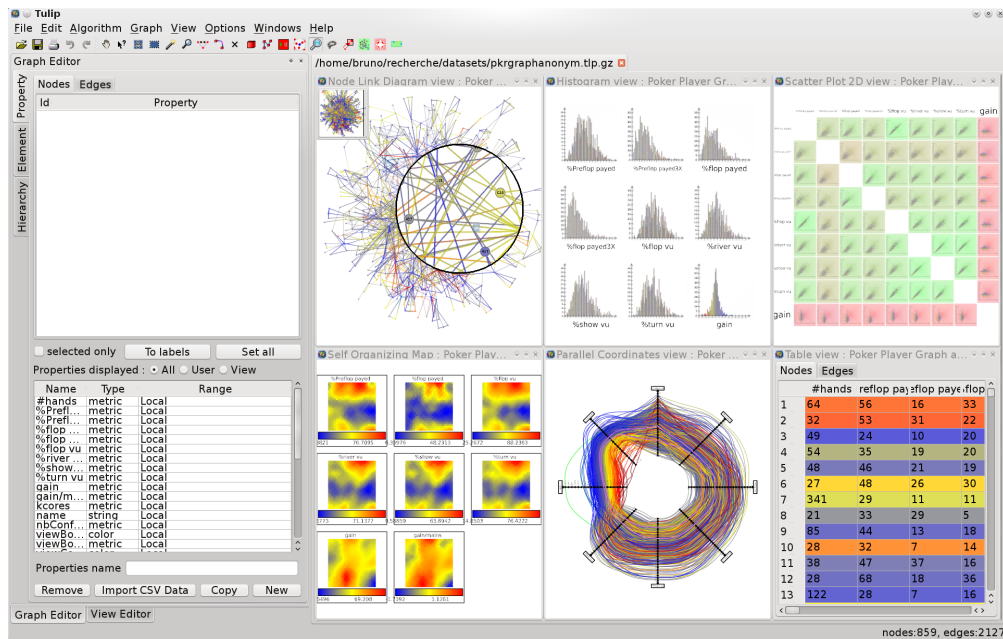


FIG. 1 – Example working session using Tulip.

2. See <http://www.eclipse.org>.

It is also very easy to add one's own routines to the whole framework. The actual methodology can actually be accessed on Tulip's website³ – see the developer's manual, user's manual and tutorials. We provide C++ templates from which one can develop its own plug-in to draw graphs, compute specific graph statistics or import data from a specific format or from a SQL database. The result of a plug-in will typically be stored in a pre-designed container called a *property*. Properties appear as a central paradigm to Tulip, allowing the framework to model and apply to a vast variety of application domains and application uses. Plug-ins can rely on each other through property invocations mechanisms. More sophisticated plug-in and interactors can be developed at the cost of dealing with QT or OpenGL or even both.

Tulip also appears as a unique framework allowing the computation and visualization of multi-level clustering algorithms. Relying on OpenGL, graph hierarchies can be interactively explored where lower level graphs appears as they gain size (Auber et al., 2007).

Recent advances with Tulip showed how GPU programming could be used to improve performance when drawing graphs or when designing edges as high polynomials curves (Béziers) (Auber et Chiricota, 2007; Lambert et al., 2010).

Soon, it will be possible to dynamically and interactively query and modify the Tulip model by using an embedded Python interpreter. The whole Tulip library (including the QT and OpenGL libraries) is mapped inside the Python library (see Fig. 2). Contrary to a C++ plugin, with Python, there is no need to restart Tulip everytime you modify your source code. The tulip Python view is perfect for developing simple script and interaction with the data model. The first end-user experiences are very positive : “Since Tulip proposed Python script my life has changed ! No need to do huge C++ development anymore !”.

3 Future Works

Tulip development is still very active. We are currently working on a version of Tulip for the web (graph visualization and interaction inside a web browser) and on the support of different types of dynamic graph (graphs which evolve over time).

Références

- Auber, D. (2003). Tulip - a huge graph visualization framework. In P. Mutzel et M. Jünger (Eds.), *Graph Drawing Software*, Mathematics and Visualization Series. Springer Verlag.
- Auber, D. et Y. Chiricota (2007). Improved efficiency of spring embedders : Taking advantage of gpu programming. In *Sixth IASTED International Conference on Visualization, Imaging, and Image Processing VIIP 2007*, Palma de Mallorca, Spain, pp. 169–175. IASTED.
- Auber, D., Y. Chiricota, M. Delest, J.-P. Domenger, P. Mary, et G. Melançon (2007). Visualisation de graphes avec tulip : exploration interactive de grandes masses de données en appui à la fouille de données et à l'extraction de connaissances. In M. Noirhomme-Fraiture et G. Venturini (Eds.), *Revue des Nouvelles Technologies de l'Information (actes EGC 2007)*, Volume 1, pp. 147–156. Cépaduès.

3. See <http://www.tulip-software.org>.

Tulip: a Scalable and Adaptive Graph Visualization Framework

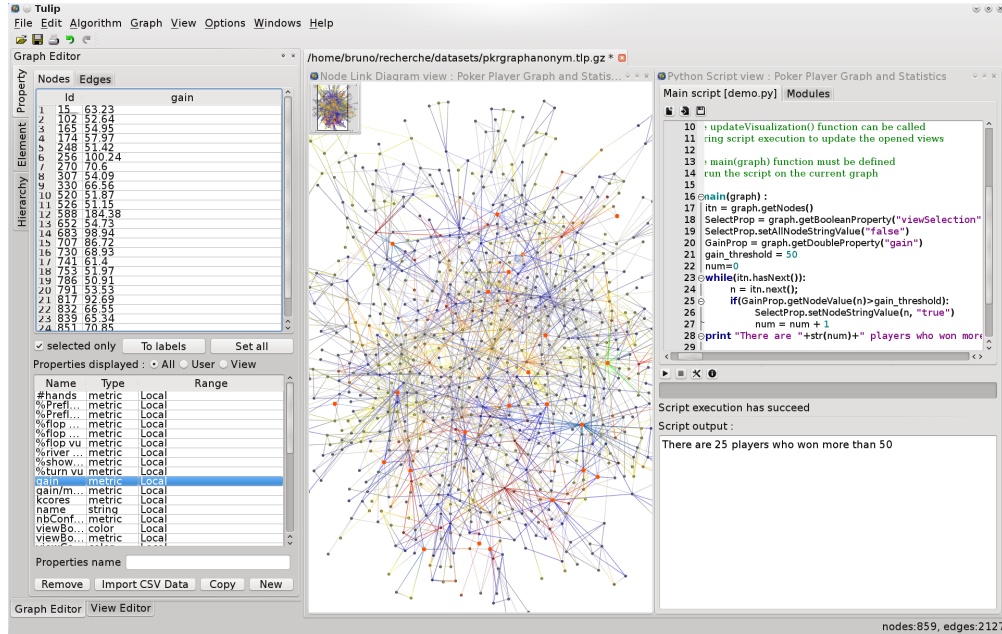


FIG. 2 – Python inside Tulip demo with the poker players example. This simple script selects the players who won more than 50. The results can be immediately visualized in the node-link diagram view.

- Auber, D. et P. Mary (2006). Mise en place d'un mécanisme de plugins en c++. *Programmation sous linux* 5(5), 74–79.
- Batagelj, V. et A. Mrvar (2002). Pajek - analysis and visualization of large networks. In *Graph Drawing : 9th International Symposium, GD 2001*, Volume 2265 of *Lecture Notes in Computer Science*, Vienna, Austria, pp. 477.
- de Nooy, W., A. Mrvar, et V. Batagelj (2004). *Exploratory Social Network Analysis with Pajek*. Cambridge University Press.
- Ellson, J., E. Gansner, L. Koutsofios, S. North, et G. Woodhull (2003). Graphviz and dynagraph - static and dynamic graph drawing tools. In M. Junger et P. Mutzel (Eds.), *Graph Drawing Software*, pp. 127–148. Springer-Verlag.
- Ellson, J., E. Gansner, L. Koutsofios, S. C. North, et G. Woodhull (2002). Graphviz - open source graph drawing tools. In *Graph Drawing : 9th International Symposium, GD 2001*, Volume 2265 of *Lecture Notes in Computer Science*, Vienna, Austria, pp. 483–484.
- Fekete, J.-D. (2004). The infovis toolkit. In *10th IEEE Symposium on Information Visualization*, Austin, TX, pp. 167–174. IEEE Press.
- Gansner, E. R. et S. C. North (2000). An open graph visualization system and its applications to software engineering. *Software Practice and Experience* 30(11), 1203–1233.

- Heer, J., S. K. Card, et J. A. Landay (2005). *prefuse : a toolkit for interactive information visualization*. In *SIGCHI conference on Human factors in computing systems*, Portland, Oregon, USA, pp. 421–430. ACM Press.
- Lambert, A., R. Bourqui, et D. Auber (2010). *Winding roads : Routing edges into bundles*. *Computer Graphics Forum* 29(3), 853–862.

Summary

La plate-forme de visualisation de graphes Tulip est maintenant riche de dix années d'expérience, tant du point de vue de son utilisation, de la conception de son architecture que de son développement logiciel. Conçue dès le départ pour manipuler de très grands graphes dans un contexte interactif, elle adopte les concepts et pratiques de l'état de l'art du génie logiciel et implémente une large palette de représentations de graphes. Destinées à être utilisées en conjonction avec des outils d'analyse de données, son architecture lui confère toute l'adaptabilité que l'on peut souhaiter en permettant à l'utilisateur d'intégrer ses propres composants d'analyse et de dessin. Les dernières versions de Tulip rendent sa mise-à-jour et l'ajout de nouvelles fonctionnalités des plus facile via la connexion aux serveurs de plug-ins.

Acquisition, Visualisation 3D et interaction pour le suivi de dissection de fibres cérébrales du cerveau humain

Barthélemy Serres*, Christophe Destrieux**, Gilles Venturini*

*Université François Rabelais de Tours, Laboratoire d'informatique,
64, av. Jean Portalis, 37200 Tours, France
{serres, venturini}@univ-tours.fr

**UMRS U930 INSERM, CNRS ERL3106, 2 bvd Tonnellé, 37000 Tours, France
destrieux@med.univ-tours.fr

Résumé. Les fibres blanches sont impliquées dans de nombreuses maladies du système nerveux central. La connaissance de ces fibres est primordiale afin de mettre en place un traitement adapté. Nous apportons la conception d'un système complet depuis l'acquisition, jusqu'à la visualisation interactive. L'acquisition est réalisée en deux temps, d'abord avec un scanner laser pour obtenir une surface 3D, puis ensuite avec un appareil photographique pour la texture. Cette dernière est ensuite plaquée sur la surface 3D pour obtenir une surface texturée. Ce processus est répété sur plusieurs étapes de la dissection en cours. Les surfaces doivent ensuite être recalées entre elles à l'aide de repères fixes. Ces surfaces 3D sont visualisées en 3D stéréoscopique. Un outil d'étiquetage interactif est proposé à l'expert neurochirurgical pour identifier les faisceaux d'intérêt. L'objectif final est la reconstruction 3D de ces faisceaux de fibres blanches identifiés sur l'ensemble des surfaces acquises.

1 Introduction

Ce travail s'inscrit dans un projet mené en collaboration avec des neurochirurgiens¹, qui s'intéressent à la constitution d'un atlas de fibres blanches pour le cerveau humain. Les fibres blanches sont des structures cérébrales reliant les différentes aires corticales. Pour cette étude, nous avons développé le protocole suivant : une acquisition IRM in-vivo sur un sujet ayant fait don de son corps est effectuée. Après son décès, l'encéphale est prélevé et préparé pour dissection selon la méthode décrite par Ludwig et Klingler (1956). Cette préparation consiste principalement en plusieurs phases de congélation et décongélation ayant pour objectif de faciliter la dissection des fibres blanches. Le chirurgien réalise alors la dissection des hémisphères, étape par étape. Ce travail de haute précision demande environ 80 heures pour un encéphale complet. Une acquisition surfacique et photographique de la pièce anatomique est effectuée à chaque étape. Après la dissection, le neurochirurgien visualise en 3D stéréoscopique les surfaces acquises et y surligne les faisceaux d'intérêt avec du matériel d'interaction. Ceux-ci seront alors reconstruits de manière précise, pour être recalés ensuite sur l'IRM in-vivo. Un atlas pourra

1. Unité Inserm U930/CNRS ERL3106, équipe "Imagerie et Cerveau"

alors être constitué avec plusieurs sujets. Nous présentons ici une méthode d'acquisition complète ainsi qu'un outil permettant à l'expert anatomiste d'effectuer interactivement ce suivi des fibres.

2 Acquisition

Dans notre protocole, les encéphales préalablement préparés sont divisés en deux parties, selon les hémisphères gauche et droit, puis fixés sur une platine grâce à de la paraffine. Lors de la dissection et à plusieurs instants clef du déroulement de celle-ci, le neurochirurgien acquiert la surface de la pièce anatomique selon deux modalités, en vue d'obtenir d'une part la géométrie de la pièce anatomique et d'autre part l'information de texture couleur.

Acquisition de surface Le choix de la méthode d'acquisition 3D est important, puisque la précision et les résultats de l'étude en dépendent directement. Depuis quelques années de nombreux travaux s'intéressent à l'utilisation de dispositifs laser d'acquisition 3D dans le domaine médical. Kovacs et al. (2007) ont travaillé sur l'acquisition de données morphologiques avec un scanner laser pour ensuite comparer ces données à celles acquises par des techniques classiques d'imagerie tomographique (CT). Wolski et al. (2009) se sont attachés à démontrer l'utilité d'un suivi de résultats du traitement de pathologies grâce à l'utilisation d'un scanner laser. C'est pourquoi nous avons choisi de mettre en œuvre un périphérique habituellement utilisé dans le contrôle qualité industriel pour sa grande précision, le scanner laser. Ce matériel¹ est constitué d'un bras articulé surmonté d'une tête à balayage laser, et permet l'acquisition de surfaces avec une précision nécessaire à l'étude (précision du bras articulé : $\pm 90\mu m$, précision de la tête laser : $\pm 50\mu m$). En effet, les partitions de surfaces identifiées par le neurochirurgien, comme étant les faisceaux d'intérêt doivent être le plus fidèle possible à la réalité anatomique. Ces modèles 3D ainsi obtenus (cf. Fig 1) sont traités géométriquement dans une application tierce². Les traitements effectués sont le filtrage des points aberrants et des triangles dégénérés, ainsi qu'une optimisation légère des triangles coplanaires. Les modèles sont ensuite exportés sous forme de fichiers contenant une liste points/facettes classique.

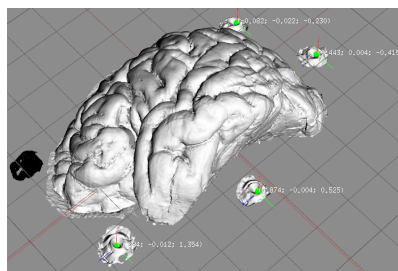


FIG. 1: Maillage résultant de l'acquisition surfacique

1. Faro Laser ScanArm, FARO Technologies, USA
2. Geomagic Studio, Geomagic Inc, Research Triangle Park, NC, USA

Acquisition de texture L'acquisition de texture est effectuée avec un appareil photographique commercial³. Un rendu photographique de la scène est en effet nécessaire pour obtenir des modèles photo-réalistes, et ainsi offrir aux experts la possibilité d'identifier aisément et de manière précise les régions d'intérêts constitués par les faisceaux de fibres. Les clichés sont acquis à haute résolution et orthogonalement par rapport à notre zone d'intérêt (cf. Fig 2). Nous effectuons un seul cliché par étape de dissection. Les structures recherchées sont principalement orientées dans le plan XY, alors que la dissection se déroule selon l'axe Z. Les conditions d'éclairage de la scène ont également été prises en compte et mises au point avec le concours d'un photographe professionnel. Nous avons dû prendre en compte les contraintes suivantes : élimination des reflets, et ombrages, répétabilité du système puisque plusieurs prises de vue à des instants différents sont nécessaires. La technique d'éclairage retenue est basée sur l'utilisation de flashes annulaires, permettant de répondre au mieux à ces contraintes.

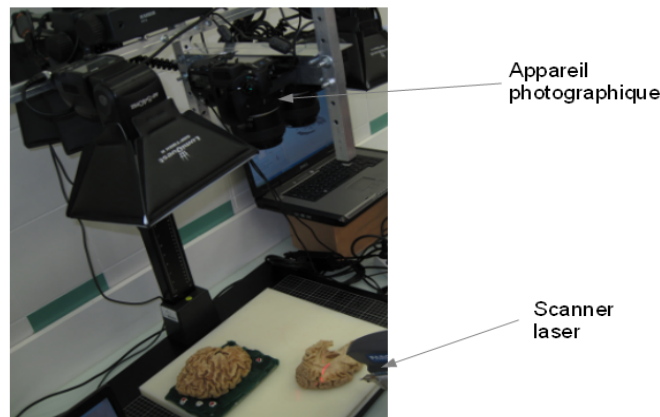


FIG. 2: Aperçu du dispositif utilisé pour les acquisitions

3 Visualisation 3D

Afin de visualiser l'étape de dissection désirée, il est nécessaire d'associer le résultat de la capture laser 3D (surface 3D) à la photographie (texture). Nous obtenons alors des surfaces texturées que nous nommons *acquisition*. Une dissection complète est donc la composition de plusieurs *acquisitions*.

Plaquage de texture Le plaquage de texture est l'étape qui consiste à associer à chaque coordonnées 3D (x,y,z) de l'objet virtuel, une coordonnée 2D de texture (u,v). Ici, le plaquage de la texture est particulier puisqu'il est dépendant de l'angle de vue (*view dependent texturing*). En effet, la texture provient d'un point de vue unique dans la scène réelle. On ne peut approximer le plaquage avec une méthode de projection orthogonale classique.

3. K20 Pentax, Pentax Corporation, Japan

On sait que le plaquage de texture peut être vu comme une application linéaire permettant d'associer à chaque vertex $v_{x,y,z,w}$ des coordonnées de texture $v_{s,r,t,q}$ avec :

$$\begin{bmatrix} s \\ r \\ t \\ q \end{bmatrix} = T_{\text{plaquage}} * \begin{bmatrix} v_x \\ v_y \\ v_z \\ v_w \end{bmatrix}$$

La méthode utilisée permet un plaquage à la manière d'un projecteur optique présentée par Everitt (2001). Nous modélisons ici le projecteur par deux matrices :

- M_{position} modélisant sa position et son orientation,
- $M_{\text{projection}}$ modélisant les propriétés intrinsèques de projection.

Ces dernières propriétés faisant intervenir le ratio d'affichage ($\frac{\text{width}}{\text{height}}$) l'angle de champs ou *Field Of View*.

On pose M_{objet} , la matrice de transformation de l'objet, et $M_{\text{ScaleAndBias}}$ la matrice permettant de passer de l'intervalle $[0; 1]$ des coordonnées de texture à $[-1; 1]$ permettant ainsi de centrer la projection en $(0; 0)$.

On a donc :

$$M_{\text{ScaleAndBias}} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On pose les réels a, e, n, f tels que :

- a définit le ratio d'affichage

$$a = \frac{\text{width}}{\text{height}}$$

- e dépend de l'angle de champs

$$e = \frac{2}{\tan(\text{FOV Angle})}$$

- n est la distance au plan de projection le plus proche désiré
- f est la distance au plan de projection le plus loin désiré

On aboutit à la définition de $M_{\text{projection}}$:

$$M_{\text{projection}} = \begin{bmatrix} \frac{e}{a} & 0.0 & 0.0 & 0.0 \\ 0.0 & e & 0.0 & 0.0 \\ 0.0 & 0.0 & \frac{n+f}{f-n} & -1.0 \\ 0.0 & 0.0 & \frac{2*f*n}{f-n} & 0.0 \end{bmatrix}$$

On obtient donc finalement la transformation T_{plaquage} qui est alors donnée par :

$$T_{\text{plaquage}} = M_{\text{ScaleAndBias}} * (M_{\text{projection}} * (M_{\text{position}} * M_{\text{objet}}))$$

Recalage entre surfaces L'étape du recalage est primordiale. En effet, les différentes surfaces ayant été acquises à des instants différents, il est nécessaire d'effectuer un recalage de ces surfaces les unes par rapport aux autres. Pour cela, nous avons fixé des repères sur le support rigide de l'hémisphère en cours de dissection. Ces repères sont matérialisés par des vis dont la tête a été usinée pour faire apparaître une demi-sphère creuse (cf. Fig 3b). Le scanner laser utilisé est muni d'une pointe palpeur et le point de contact est une sphère de diamètre connu (3 mm) à l'extrémité. Ainsi, avant chaque acquisition surfacique, chaque repère est *palpé*. Les coordonnées des repères étant connus nous pouvons maintenant les utiliser comme base fixe pour effectuer le recalage.

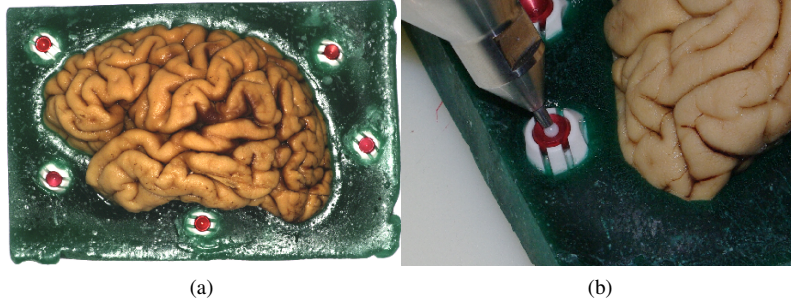


FIG. 3: Disposition (a) et palpation (b) des repères fixes

Notre recalage 3D entre surfaces est dit "rigide" puisque tous les points subissent la même transformation et il n'y a pas de changement d'échelle.

On pose $r_{i,n}$ le repère i de la surface n , avec $n > 0$.

La méthode se déroule en deux étapes :

- Mise en correspondance des points de repères, constitution des couples $(r_{*,n} ; r_{*,n-1})$
- Évaluation de la transformation $r_{*,n} \mapsto r_{*,n-1}$ (1 Rotation et 1 Translation)

Pour l'évaluation de la transformation, on utilise dans un premier temps 3 repères sur chaque surfaces n et $n-1$. Nous "normalisons" les points de la manière suivante :

$$r_{i,n}normalise = r_{i,n} - \overline{r_n}$$

avec :

$$\overline{r_n} = \frac{1}{3} \sum_{i=1}^3 r_{i,n}$$

Voici la méthode qui nous permet de trouver la rotation entre les deux triplets de points repères :

1. Choix d'un point comme étant l'origine, soit $r_{0,n}$
2. Construction de l'axe X :

$$\vec{x} = \frac{r_{1,n} - r_{0,n}}{|r_{1,n} - r_{0,n}|}$$

3. Construction de l'axe Y

$$\vec{y} = (r_{2,n} - r_{0,n}) - [(r_{2,n} - r_{0,n}) \cdot \vec{x}] \vec{x}$$

$$\vec{y} = \frac{\vec{y}}{|\vec{y}|}$$

4. Construction de l'axe Z

$$\vec{z} = \vec{x} \otimes \vec{y}$$

5. Construction des matrices $R_{r_{*,n}}$ et $R_{r_{*,n-1}}$ représentant les matrices locales de rotation

$$R_{r_{*,n}} = [r_{0,n}; r_{1,n}; r_{2,n}]$$

$$R_{r_{*,n-1}} = [r_{0,n-1}; r_{1,n-1}; r_{2,n-1}]$$

6. Construction de la matrice R transformant les $r_{*,n}$ en $r_{*,n-1}$

$$R * R_{r_{*,n}} = R_{r_{*,n-1}}$$

7. Résolution de la matrice de rotation

$$R_{r_{*,n} \rightarrow r_{*,n-1}} = R_{r_n} * R_{r_{*,n-1}}^T$$

8. Calcul du vecteur de translation

$$\forall i, T_{r_{*,n} \rightarrow r_{*,n-1}} = r_{i,n} - R * r_{i,n-1}$$

Cette méthode directe requiert bien entendu que l'on aie correspondance entre la configuration spatiale des 3 points de la surface n de "départ" avec celle des 3 points de la surface $n - 1$ d'"arrivée".

4 Interactions en Réalité Virtuelle

Les zones d'intérêt sont les faisceaux identifiés par le neurochirurgien lors de la phase de *labellisation*. Cette procédure est exécutée sur chaque *acquisition* et est divisée en deux phases. L'expert surligne des zones du maillage global grâce à un outil de sélection de triangles. Dès lors que la sélection est validée, l'expert lui attribue une classe.

Sélection de zones Une zone de sélection est constituée d'un sous ensemble de triangles de la surface 3D. Cet ensemble pouvant être disjoint. Les triangles formant cette sélection sont surlignés. L'outil de sélection de triangles intègre des opérations de l'arithmétique ensembliste comme :

- Union de plusieurs sélections,
- Complément d'une sélection,
- Inversion de la sélection

Ces fonctionnalités permettent une flexibilité nécessaire dans ce processus d'identification et de sélection. Actuellement, ces sélections sont possibles avec la souris. L'utilisateur doit donc d'abord se positionner dans le point de vue lui permettant d'effectuer sa sélection. Nous avons 3 types de sélections :

- une sélection rectangulaire,
- une sélection selon un segment,
- une sélection libre

Par la suite, d'autres périphériques peuvent être ajoutés pour apporter une sélection 3D, c'est à dire par l'intermédiaire d'un pointeur 3D. Ce pointeur 3D "suivrait" la surface à annoter.

Annotation des zones d'intérêts Après la phase de sélection de la zone d'intérêt, vient le processus d'affectation de classe. Une zone d'intérêt est une sélection qui a été validée et à laquelle une classe a été affectée. L'appartenance à une classe i est traduite par une couleur C_i identifiant de manière unique cette classe. Chaque triangle constituant la zone est étiqueté. Un triangle ne peut appartenir qu'à une seule zone.

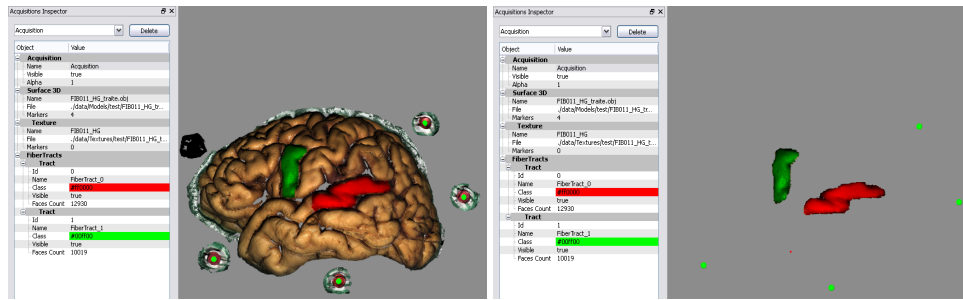


FIG. 4: Zones d'intérêts annotées

Lors de cette phase d'annotation interactive, il est possible de masquer ou d'afficher les sous-ensembles de triangles d'une ou de plusieurs classes identifiées, afin d'améliorer la lisibilité de la scène (cf. Fig 4).

Navigation inter-surfaces Nous avons choisi de pouvoir visualiser plusieurs surfaces ensemble. Il est nécessaire de différencier la surface courante, sur laquelle l'expert sélectionne les zones d'intérêt, des autres surfaces correspondant aux autres étapes de dissection. Pour cela, nous avons choisi de dissocier le rendu visuel des surfaces affichées. Seule la surface courante est affichée en couleur, alors que les surfaces voisines sont mises en retrait en désaturant la texture et en ajoutant de la transparence (cf. Fig 5). Une problématique importante à laquelle nous sommes confronté est la taille de nos surfaces ainsi que leur nombre. En effet, une dissection complète d'un encéphale représente plus de 50 objets (maillages 3D) constitué de 600 000 triangles en moyenne. Il s'agit de permettre une navigation fluide entre les surfaces, tout en permettant d'accéder à l'intégralité des données. Pour atteindre cet objectif, nous envisageons un chargement et déchargement dynamique des *acquisitions*.

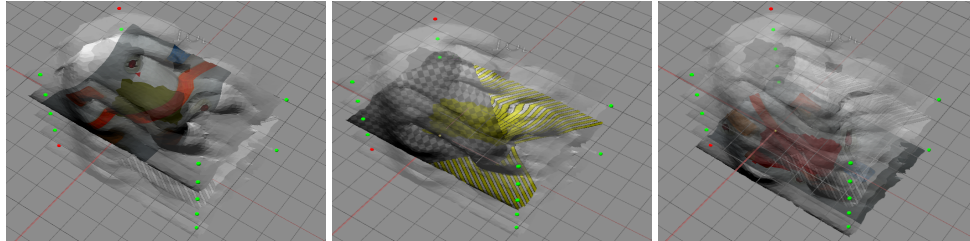


FIG. 5: Navigation multi-surfaces

Camera 3D L'utilisateur interagit avec les éléments de la scène grâce à un dispositif de navigation 3D type souris 3D ainsi qu'une souris 2D classique. La navigation 3D est contrainte par rotation autour d'un point pivot. La souris 3D possède bien six degrés de liberté, mais nous avons choisi ce mode afin de garantir un temps d'apprentissage court ainsi qu'une manipulation intuitive des éléments. Cependant, l'utilisateur conserve la possibilité, grâce au dispositif de pointage 2D, de zoomer et dé-zoommer pour se focaliser sur son point de pivot.

Stéréoscopie La visualisation des différentes surfaces texturées, s'effectue en stéréoscopie. La visualisation stéréoscopique permet de bien percevoir la profondeur d'une scène 3D et de favoriser ainsi l'immersion de l'utilisateur dans l'environnement. Nous utilisons du matériel stéréoscopique grand public constitué d'un émetteur infra-rouge, de paires de lunettes actives à obturation ainsi qu'un périphérique d'affichage. Cet écran ou vidéo projecteur doit être capable de supporter une fréquence d'affichage de 120Hz afin de garantir au moins une fréquence de 60Hz pour chaque œil. Le rendu stéréoscopique se construit ensuite en générant deux points de vue correspondant aux points de vue des yeux gauche et droit. Il existe plusieurs techniques pour la visualisation en stéréoscopie comme l'affichage entrelacé, l'utilisation de filtres polarisant. La technique qui donne une meilleure perception est celle basée *page flipping*, c'est à dire un affichage séquentiel des images gauche et droite.

5 Optimisations

Afin de garantir un rendu temps réel (>20 images par seconde) de la manipulation des objets de la scène, nous nous sommes attachés à optimiser l'organisation des données. En effet, de nombreuses requêtes vont être effectuées sur la géométrie (lancés de rayons, recherches de plus proches voisins, etc...).

Partitionnement spatial Les objets géométriques obtenus doivent être de bonne définition, ce qui porte le nombre de triangles dans notre cas de 650 000 à 700 000 pour une seule surface. Nous avons donc utilisé une structure de partitionnement de l'espace introduite par Samet (1990), l'*octree*. L'*octree* est une structure hiérarchique récursive de cellules cubiques. Chacune de ses cellules est soit terminale, soit non terminale. Dans le cas d'une cellule non terminale, celle-ci possède forcément huit fils cubiques la divisant équitablement. Il est composé d'une cellule racine étant la boîte englobante cubique minimale. Une profondeur de subdivision est définie avant la construction. Une telle structure permet lors de calculs d'intersection

de rayons, depuis la racine jusqu'aux cellules feuilles, d'optimiser la traversée du maillage en profitant des zones "vides".

Indexation des points de repères Avant la phase de recalage basée sur les points repères, il est nécessaire d'établir les correspondances entre les repères des surfaces à recaler. Cette étape pourrait dans un premier temps être réalisée manuellement, mais comme ce processus doit être répété sur les $N-1$ couples de surfaces à recaler, il est intéressant d'apporter une méthode automatique. L'algorithme utilisé ici est donc basé sur l'Iterative Closest Point (ICP) introduit par Besl et McKay (1992). A chaque itération de l'algorithme des correspondances sont donc établies entre les point de repères de la surface $N-1$ et ceux de la surface N . Le critère choisi est la distance euclidienne entre deux points appartenant N et de $N-1$. Nous cherchons donc à minimiser l'écart entre chaque couple de points repères (1). L'algorithme itère jusqu'à ce que ce que l'erreur soit inférieure à un ϵ choisi. Avec m repères, P et P' ensembles des repères 3D sur les deux surfaces, on obtient :

$$\text{Minimize} \sum_{i=1}^m ||P_i - P'_i||^2 \quad (1)$$

Lors de la phase de mise en correspondance des repères, les calculs les plus coûteux mis en jeu sont des recherches de plus proches voisins. C'est pourquoi nous avons choisi d'adjoindre au maillage une structure proposée par Moore (1991) efficace pour ce type de problème, le kd-tree. Le kd-tree permet une "indexation" des points apportant une accélération pour la recherche des plus proches voisins.

6 Résultats

Recalage entre surfaces Le système de recalage permet actuellement de repositionner les surfaces par rapport à une référence choisie par l'utilisateur. Pour cela, nous avons acquis une même scène dans deux référentiels différents. Après recalage, nous avons mesuré les écarts entre les points repères de deux surfaces. Voici les résultats que nous avons obtenus (cf. Tab. 1). Il est important de noter que le recalage précis des surfaces 3D permet par la suite une

	écart min	écart max	écart moyen
surface0/surface1	0.142	0.266	0.2

TAB. 1: Écarts de recalage mesurés (en mm)

reconstruction plus fidèle des faisceaux de fibres. Cependant, la modalité d'acquisition, en IRM de diffusion, fournit des images 3D d'une résolution spatiale bien inférieure (≈ 1.5 à 2mm pour un voxel), la précision de notre système est donc très bonne.

Visualisation 3D Le système actuel permet une sélection interactive au niveau triangle avec un affichage en stéréoscopie en temps réel. Il permet l'affichage simultané de plusieurs surfaces. Voici un tableau récapitulant les performances obtenues, en fonction du nombre de surfaces en présence (cf. Fig 6). On note que les performances ne s'écraquent pas avec la montée

en charge et que l'on reste au dessus du seuil d'interaction². Notre système de visualisation permet donc de visualiser un volume de données important tout en conservant bien la notion d'interactivité temps réel.

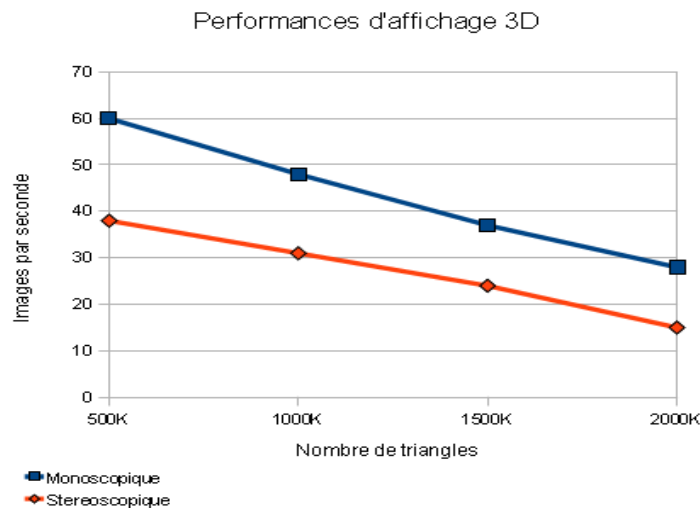


FIG. 6: Évolution des performances d'affichage avec la charge (en image/s)

7 Conclusion et perspectives

Dans cet article, nous avons décrit les étapes des problématiques auxquels nous avons répondu par nos propositions depuis l'acquisition des données jusqu'à leur visualisation en réalité virtuelle. Ceci fait l'objet depuis un an du développement d'une plate-forme de visualisation élaborée en étroite collaboration avec l'équipe d'experts médicaux du projet, dont leurs retours permettent d'adapter l'ergonomie de l'outil.

Concernant l'interaction avec l'utilisateur, nous travaillons actuellement à l'intégration d'un périphérique de retour haptique dans la plate-forme afin d'apporter une dimension sensitive supplémentaire pour l'extraction de connaissance à partir de la structure géométrique. Ce périphérique permet également de positionner de manière intuitive un pointeur en 3D, ce qui nous paraît important notamment pour le travail en stéréoscopie.

Concernant le recalage 3D des *acquisitions* qui s'effectue actuellement sur 3 points de repères, nous travaillons sur la prise en compte de plus de points repères. Nous travaillons également à l'élaboration d'une méthode de recalage non plus basée uniquement repères, mais plutôt basée sur un sous-nuage de points communs aux différentes surfaces que l'on souhaite recalculer. Nous comparerons ensuite la qualité des différents résultats de recalage obtenus.

2. Tests réalisés avec un processeur Intel Xeon@2.00GHz et un GPU NVidia Quadro FX4600

Enfin, la suite de nos travaux portera principalement sur la reconstruction des faisceaux de fibres annotés dans l'optique de comparer les résultats obtenus à ceux issus d'images IRM de diffusion et reconstruits par tractographie classique. Kao et Tarng (2006) ont travaillé sur les problématiques de recalages de surfaces entre différentes modalités d'acquisition comme l'image CT avec des surfaces issues de scans laser, alors que Lomonosov et al. (2006) ont présenté un travail sur le recalage entre surfaces, en utilisant une approche basée sur des algorithmes évolutionnaires.

Références

- Besl, P. J. et N. D. McKay (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2), 239–256.
- Everitt, C. (2001). Projective texture mapping. Technical report, NVidia Corp.
- Kao, J. et Y. Tarng (2006). The registration of ct image to the patient head by using an automated laser surface scanning system—a phantom study. *Computer Methods and Programs in Biomedicine* 83(1), 1 – 11.
- Kovacs, L., M. Eder, R. Hollweck, A. Zimmermann, M. Settles, A. Schneider, M. Endlich, A. Mueller, K. Schwenzer-Zimmerer, N. A. Papadopoulos, et E. Biemer (2007). Comparison between breast volume measurement using 3d surface imaging and classical techniques. *The Breast* 16(2), 137 – 145.
- Lomonosov, E., D. Chetverikov, et A. Ekárt (2006). Pre-registration of arbitrarily oriented 3d surfaces using a genetic algorithm. *Pattern Recognition Letters* 27(11), 1201 – 1208.
- Evolutionary Computer Vision and Image Understanding.
- Ludwig, E. et L. Klingler (1956). The inner structure of the brain. *Atlas cerebri humani*.
- Moore, A. (1991). An introductory tutorial on kd-trees. Technical Report 209, Computer Laboratory, University of Cambridge, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- Samet, H. (1990). Applications of spatial data structures : Computer graphics, image processing, and gis.
- Wolski, M., C. Brack, et I. Kessel (2009). Utility of 3d surface laser scanning in brachytherapy of lip cancer. *International Journal of Radiation Oncology*Biophysics* 75(3, Supplement 1), S604 – S604. Proceedings of the American Society for Radiation Oncology 51st Annual Meeting, 51st Annual Meeting of the American Society for Radiation Oncology.

Summary

White fibers are part of brain's white matter. White fibers are involved in several diseases of the central nervous system. Knowledge of these fibers is necessary to setup an effective treatment. We have designed a full featured system from surface and texture capture, to interactive visualization of brain anatomy. Capture is realized in two passes, first we use a 3D laser scanner to get the 3D surface, and then we use a digital camera to get the texture. The resulting picture is used to texture the 3D surface. This process is repeated on every step of the brain

Acquisition, Visualisation 3D, Interaction pour la reconstruction de fibres cérébrales

dissection. Then, 3D surfaces are displayed in stereoscopy and an interactive tool is provided to the neurosurgeon expert, in order to identify tracts of interest. The final goal is to rebuild these identified white fibers tracts using the whole surface set.

Librairie Open Source de Visualisation Interactive et Immersive

Nicolas Greffard*, Fabien Picarougne*
Pascale Kuntz*

*Equipe COD - LINA - Polytech'Nantes, rue Christian Pauc
BP50609 F-44306 Nantes Cedex 3
{prénom.nom}@univ-nantes.fr,
<http://http://www.polytech.univ-nantes.fr/COD/>

Résumé. Cet article présente les premières briques d'un nouveau framework de visualisation interactive de données complexes et volumineuses. Cet outil permet de faciliter le développement d'applications de visualisation dans un contexte immersif en prenant en charge divers périphériques d'interaction (souris 3D, wiimote, caméra de tracking) et différents modes de représentation (stéréoscopique, auto-stéréoscopique, dôme de visualisation). En se focalisant sur la facilité d'utilisation, ce framework se comporte comme une boîte à outils proposant des mécanismes simplifiés permettant de gérer l'affichage des données tri-dimensionnelles, l'interface utilisateur et les interactions avec ce dernier. Nous présentons également différentes applications développées sur la base de ce framework permettant de visualiser avec des techniques de rendu complémentaires (projectif, volumique) des données volumineuses (comprenant plusieurs millions de facette) mais également dynamiques, en temps réel.

1 Introduction

La visualisation d'information est devenu un outil de plus en plus répandu et utilisé dans de nombreux domaines de recherche. Ce mode de restitution de l'information utilise les capacités du cerveau humain, fortement impliquées dans le système de perception visuel (Ware, 2000). Il permet au sujet qui analyse des données de les interpréter plus efficacement que lors d'une exploitation textuelle (Dix et al., 2003). À titre d'exemple, dans le cas de l'étude des réseaux sociaux, la visualisation, en sus de permettre une restitution facilement appréhendable d'un réseau, permet d'utiliser le système cognitif humain comme outil de reconnaissance de forme. Dans un autre domaine, en médecine, la visualisation est également devenue indispensable ; l'utilisation de scanners permet de visualiser l'intérieur du corps humain sans procédure chirurgicale et invasive pour le patient humain. Les mêmes techniques d'exploration de données sont utilisées par les biologistes en cognition animale, qui cherchent à analyser les mécanismes et principes engendrant des dynamiques complexes et des réseaux d'interactions dans les sociétés animales. Un élément clé est l'exploration des comportements animaux dans leur habitat (Valverde et al., 2009). Cependant, ces derniers sont souvent difficilement accessibles in vivo

sans risquer d'endommager leurs structures. Il est donc primordial d'avoir recours à des procédés d'acquisition non-destructifs, parfois complexes et indirects (imagerie aux rayons X, relevés de position par stéréoscopie), afin de préserver son intégrité.

Si de nombreux outils ont été proposés ces dernières années dans la communauté de visualisation de l'information, mais aussi dans celle de l'extraction de connaissances dans les données, peu d'entre eux utilisent la 3D immersive comme support de visualisation interactif. Les travaux pionniers de P. Bourke (Bourke, 1999) ont cependant ouvert des pistes très prometteuses et la popularisation croissante des outils 3D rendent cette voie de recherche de plus en plus attractive.

Dans ce manuscrit, nous présentons les premières briques d'un framework de visualisation interactive et immersive. Ces briques ont été testées expérimentalement dans plusieurs projets interdisciplinaires.

La suite de ce papier est organisée de la façon suivante, nous présentons tout d'abord dans la section 2, un état de l'art de différentes solutions permettant d'aider au développement d'un logiciel de visualisation. Nous présentons ensuite dans la section 3 les techniques et méthodes de rendu que nous avons actuellement implémentées dans notre framework. Cette description s'accompagnera d'exemples d'applications illustrant l'intérêt de chaque type de représentation. Dans la section 4, nous présenterons les environnements de restitution et d'interaction pris en charge par notre framework. Et, nous concluons ce papier dans la section 5 en introduisant les perspectives émergentes de ces travaux.

2 Solutions existantes

Dans le cadre d'un projet d'implémentation d'application de visualisation, la sélection d'un logiciel ou d'une librairie de rendu graphique est une étape primordiale qui peut s'avérer laborieuse. Nous présentons ici les solutions les plus couramment employées dans la littérature. Le spectre des domaines d'application de logiciels de visualisation étant très étendu, nous nous focalisons sur les solutions permettant de visualiser les types de données pris en charge par notre librairie.

Application L'imagerie médicale est un moteur de la visualisation d'information. Les images produites par les différents procédés d'acquisition (IRM, échographie, etc.) sont au format standard DICOM. Il existe de nombreuses solutions logicielles disponibles pour tous les types de systèmes d'exploitation permettant de visualiser ces données. Cependant, ces logiciels agissent avant tout comme une interface avec le matériel d'acquisition et ne permettent pas une visualisation permettant d'analyser finement les images générées avec par exemple une restitution volumique des données acquises. Ceux qui le permettent, comme par exemple *VoluMedic* ou *VTK* emploient généralement un algorithme de lancer de rayons (ray tracing ou ray casting). Ces deux solutions produisent des images d'une grande qualité, mais leur complexité algorithmique les rend inutilisables dans une application en temps réel. De plus, les logiciels qui implémentent cette fonctionnalité sont souvent propriétaires et coûteux, ce qui peut les rendre

inaccessibles. Nous pouvons toutefois souligner l'existence d'une solution alternative gratuite, *Osirix*, qui permet entre autres d'effectuer ce type d'opérations. La principale limite de ces solutions est de ne permettre de visualiser qu'un unique volume de données et d'interdire l'ajout d'autres objets visuels 3D permettant par exemple d'annoter la visualisation. Portée par la croissance explosive de Facebook, et des autres réseaux sociaux, la visualisation de réseaux de relations est un autre domaine de visualisation en plein essor. Il existe un grand nombre de logiciels permettant de visualiser ce type de données. *Visone* (Brandes et Wagner, 2003) ou encore *TouchGraph* en sont des exemples qui permettent très facilement d'obtenir une restitution visuelle d'un réseau sous forme de graphe. Une des références en la matière dans la communauté francophone est *Tulip* (Auber, 2003) qui offre de nombreuses méthodes de visualisation de réseaux, que ce soit sous forme de graphes ou de matrices. Cet outil intègre également de nombreux algorithmes spécifiques à l'étude de données relationnelles. Cependant, il ne supporte pas à ce jour -tout comme les applications analogues citées précédemment- de mode de projection 3D stéréoscopique ou immersif et se limite à un affichage 2D ou 2,5D.

Outre les données volumiques et les données relationnelles beaucoup d'autres types de données sont couramment visualisés. Par exemple les SIG (Système d'Information Géographique) visualisent l'information en superposant des plans de visualisation sur des données géographiques. A chaque type de données correspondent plusieurs applications. Cependant ces applications sont le plus souvent très spécifiques et répondent, là encore, uniquement à un problème particulier. Elles ne sont donc pas nécessairement facilement réutilisable.

Librairies Il existe également des briques logicielles aidant au développement d'une application dédiée. Les librairies provenant de l'industrie vidéoludique, plus communément appelées moteurs 3D offrent une grande variété de fonctionnalités de rendu 3D. Sans nous attarder sur les plus grands noms tel que le *CryEngine* dont le prix est un obstacle pour de nombreuses structures, nous pouvons mentionner *Ogre3D*, *Irrlicht* ou encore *OpenSceneGraph* qui sont libres et gratuits. Ces librairies se présentent comme une surcouche OpenGL, abstrayant plus ou moins de mécanismes à l'utilisateur. Cependant, celles-ci sont orientées vers le rendu 3D pour les jeux vidéos et proposent ainsi une architecture adaptée sous forme de scène. Cette architecture peut cependant s'avérer trop contraignante dans le cadre d'une visualisation scientifique. La solution proposée dans cet article se rapproche de *Visualization Library* tout en offrant plus de souplesse en terme de choix de périphériques d'affichage et d'interactions.

3 Types de données visualisés et rendu graphique

Cette section présente les différents types de données supportés par notre librairie ainsi que les formats qui leurs sont associés.

3.1 Données de type mesh 3D

Les techniques de rendu basées sur les surfaces sont largement répandues. Elles consistent, pour la plupart, à extraire une approximation polygonale d'une iso-surface. Cette surface crée une couverture de la structure visualisée et permet de dessiner une surface 3D dans un champ de données volumétriques correspondant aux points ayant une valeur d'intensité spécifique. Ce

type de données est très courant en représentation 3D et notre framework permet d'en charger de très gros volumes ; de l'ordre de la dizaine de millions de facettes en stéréoscopie tout en gardant une qualité graphique correcte et gérant un modèle d'éclairage de Phong (par pixel) (Phong, 1975). De plus, notre librairie effectue la restitution de ce type de données en temps réel, autorisant ainsi une interaction directe avec l'utilisateur. Le chargement et l'indexation des facettes sont optimisés de façon à ce que l'on puisse, par exemple, connaître très rapidement les facettes partageant des vertex communs. Ceci permet, de produire une navigation qui coïncide avec la surface de la structure visualisée.

Les capacités de rendu 3D de notre librairie ont pu être testées dans le cadre du projet MESO-MORPH en collaboration avec le Centre de Recherche sur la Cognition Animale de l'université de Toulouse. Une application d'extraction de données volumétriques -sous forme d'une série d'images DICOM décrivant des nids de termites tomographiés- a été développée, NestMiner (Picarougne, 2009). La qualité de la restitution est en grande partie déterminée par le choix de la valeur d'intensité de l'iso-surface. Il n'existe pas, à notre connaissance, de méthode rapide permettant de sélectionner cette valeur de manière automatique. Le choix du seuil d'intensité utilisé pour construire cette surface a donc été laissé à l'utilisateur.

Une fois la surface obtenue, il est nécessaire de la transformer de façon à ce qu'elle soit interprétable par la carte graphique. La méthode la plus répandue se sert de polygones connectés. En effet, les cartes graphiques récentes permettent de restituer des surfaces de polygones très efficacement. Autrefois breveté, mais désormais libre de droit, l'algorithme le plus connu d'approximation polygonale d'une surface est appelé "*Marching Cubes*" (Lorensen et Cline, 1987). NestMiner en utilise une variante corrigeant certains problèmes de cohérences topologiques observés dans certains cas, le "*Marching Tetrahedra*" (Doi et Koide, 1991). Les polygones ainsi produits peuvent être enregistré sous forme de fichier au format mesh. Une restitution de la visualisation obtenue par cette application est illustrée sur la Figure 3

3.2 Données de Type Graphe

Graphes statiques Le format mesh est de bas niveau ; il ne permet que de lister de manière exhaustive la liste des polygones composant une structure. Il n'est donc pas adapté pour décrire la représentation graphique de données structurées de façon plus abstraite. Nous avons ajouté dans notre librairie la possibilité d'interpréter des formats de plus haut niveau abstrayant la représentation graphique des données. Les données structurées sous forme de graphes par exemple, apparaissent dans de nombreux domaines de recherche comme l'étude des réseaux informatiques ou les réseaux sociaux pour ne citer que ceux-ci. Notre librairie supporte donc un format d'entrée décrivant les ensembles E et V dénotant un graphe $G = (V, E)$ avec V un ensemble d'éléments et E un ensemble de liens entre les éléments de V . Celui-ci est ensuite interprété afin de tracer une représentation du graphe G dans un espace euclidien tridimensionnel virtuel. Notons que des informations sémantiques peuvent être associées à chacun des éléments de V .

L'utilisation de notre librairie pour la visualisation de graphes permet également de tirer parti d'une de ses fonctionnalités essentielles : la projection 3D stéréoscopique. En effet, une des problématiques majeures de la visualisation de graphes provient du besoin de visualiser les données à différents niveaux de granularité afin d'interpréter la structure du graphe. Souvent très dense, le tracé d'un graphe peut ainsi être difficilement interprétable. Différentes méthodes

ont été proposées afin de pallier à ce problème. Certaines études récentes (Ware et Mitchell, 2008) semblent indiquer que la 3D stéréoscopique offre une plus value intéressante dans le cadre de la visualisation de graphes.

Graphes dynamiques Après s'être concentrés sur les problématiques liées au passage à l'échelle, différents domaines de recherche s'intéressent désormais aux données dynamiques qui retracent des évolutions temporelles. Cette tendance se vérifie notamment dans le domaine de l'étude des graphes (Moody et al., 2005).

Nous pouvons distinguer deux classes d'algorithmes permettant d'appréhender les données dynamiques : les algorithmes *offline* qui supposent que nous disposons de l'intégralité des données ; et *online*, correspondant aux données se présentant sous forme de flux. Le problème principal est alors le placement de ces objets dont certains vont apparaître ou disparaître avec le temps. Le cas *online* est plus délicat à prendre en charge car il faut s'assurer que les "données futures" puissent être insérées dans l'espace de visualisation de la manière la plus fluide possible. Outre les caractéristiques esthétiques telles que la limitation de l'occlusion ou l'occupation optimale de l'espace, il faut restreindre les disparités visuelles entre deux états consécutifs du système. En effet, cette disparité visuelle peut entraîner une modification de la carte mentale -représentation mentale que se fait l'utilisateur de la répartition des objets dans l'espace (Misue et al., 1995)- de l'utilisateur. Si celle-ci est sujette à de nombreuses perturbations le coût cognitif du réapprentissage peut devenir trop important et peut entraîner un abandon du processus.

Afin de limiter les perturbations de la carte mentale, notre librairie facilite la mise à jour des changements dans la visualisation de manière continue. Cette approche est différente de l'approche discrète qui segmente l'évolution de la visualisation et produit une représentation visuelle pour chaque segment de temps ; ce qui peut ainsi entraîner de fortes perturbations de la carte mentale en cas de changements significatifs de la structure globale de la visualisation entre deux segments. L'utilisateur est ainsi libre de s'appuyer sur cette capacité de mise à jour continue des données afin d'implémenter l'algorithme de placement de son choix.

Notre librairie offre également la possibilité d'assigner des actions à chaque élément visualisé. Il est par exemple possible de cliquer sur un objet afin de consulter le contenu qui lui est éventuellement attribué. A l'heure actuelle, cette fonctionnalité est directement intégrée dans le code, mais son implémentation sous forme de script est à l'étude.

A l'heure actuelle, deux applications distinctes exploitent cette partie de la librairie. La première est une application de visualisation de réseaux sociaux dynamiques modélisés sous forme de graphes (Greffard et al., 2010). La topologie du graphe est une dimension primordiale dans ce type d'études puisque la proximité d'un certain nombre de nœuds peut être directement interprétée comme l'appartenance à un même groupe social. L'algorithme de placement de force et ressorts est donc particulièrement adapté puisque la proximité Euclidienne encode directement la proximité sociale. Une variante de la version incrémentale de cet algorithme (McCrickard et Kehoe, 1997) est utilisée dans cette application. Cet algorithme repose sur la simulation en temps réel d'un système de forces inspirées du domaine de la Physique. Plus précisément, les objets visualisés sont assimilés à des particules électriques chargées de même polarité. Ainsi, l'application de la répulsion de Coulomb (Equation 1) tend à éloigner chaque

Librairie de visualisation

objet des autres objets.

$$F = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2 (p_1 - p_2)}{|p_1 - p_2|^3} \quad (1)$$

où p_1 et p_2 sont les coordonnées de deux objets et q_1 et q_2 les "charges électriques" de ces objets (particules) et où ϵ_0 est la constante électrique.

Il est ensuite possible d'associer certains objets entre eux via l'application de l'attraction de Hooke (Equation 2) qui tend à maintenir les objets liés à une distance définie.

$$F = -k \left(\frac{(|L| - R)L}{|L|} \right) \text{ avec } L = p_2 - p_1 \quad (2)$$

où p_1 et p_2 sont les coordonnées de deux objets et k est la constante de raideur et R la longueur à vide de la liaison.

Le système est ainsi simulé et converge vers une solution qui tend à minimiser la somme globale des forces engendrées par le système. Ce type d'algorithme de placement produit une approximation physique qui peut être facilement interprétée par un humain et dont les objets sont généralement bien distribués dans l'espace. Si un objet o apparait ou disparaît, il est retiré ou ajouté au système (par défaut, à une position aléatoire) suite à quoi l'ensemble des forces est de nouveau appliqué ce qui entraîne une mise à jour du placement des objets qui sont sujets à des forces émanant de o . Ces forces étant petites par rapport à la somme des forces absolues composant le graphe, leur influence est surtout perceptible au voisinage de o , ce qui facilite la visualisation de l'évolution du graphe à différentes granularités. Notons que les forces sont bornées afin d'éviter un déplacement infini dans l'espace virtuel.

La deuxième application exploitant les capacités de visualisation de données sous forme de graphes de notre librairie est une application de suivi visuel des différentes étapes d'un algorithme de classification d'une bibliothèque d'images (Bruneau et al., 2010). Dans ce contexte, les objets projetés sont des aperçus des images composant les différents clusters. De plus, ces clusters sont liés en fonction de leur similarité. Cet outil a été développé pour monitorer les images transitant dans un réseau. Ainsi, les données entrantes sont sous la forme d'un flux, résultat d'un algorithme de clustering qui agit en temps réel et qui offre la possibilité à l'utilisateur de modifier la composition des clusters de manière interactive.

Ces deux applications sont illustrées par la Figure 1.

3.3 Rendu volumique

Notre librairie offre également une approche de restitution directe de volumes en implémentant une méthode appelée *Texture mapping*. Cette méthode se base sur des coupes 2D ou 3D texturées, combinées en utilisant un opérateur de mélange (Kniss et al., 2006). Les données volumiques sont ainsi rendues par le biais de polygones parallèles alignés dans le sens de la

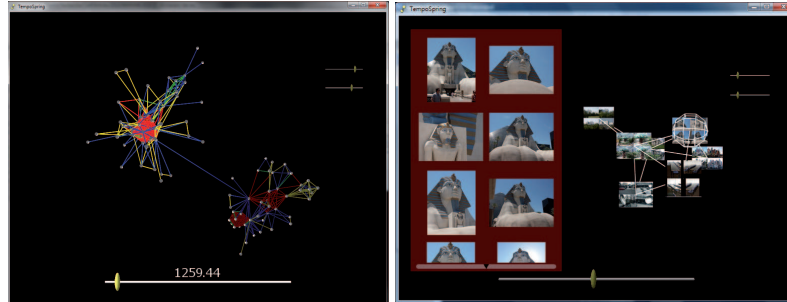


FIG. 1 – Exemple d'applications réalisées en utilisant notre librairie. (Gauche) Visualisation d'un réseau social dynamique. Les données utilisées pour construire ce graphe ont la particularité d'évoluer dans le temps. Le slider situé en bas de l'image permet de parcourir cette dimension temporelle afin de pouvoir visualiser l'état du graphe à différents pas de temps. (Droite) Visualisation de l'évolution d'un algorithme de classification d'images. Dans les deux cas, les sliders situés à droite permettent de régir les forces de l'algorithme de placement.

vue. Une texture correspondant à l'intersection entre les données volumiques et le polygone leur ait ensuite appliqué. Un moteur de rendu classique est ensuite employé afin d'afficher ces polygones de l'arrière plan vers le premier plan par rapport au point de vue de l'utilisateur afin de permettre l'utilisation de la transparence, difficile à implémenter pour la visualisation classique de mesh 3D. Cette méthode est illustrée par la Figure 2.

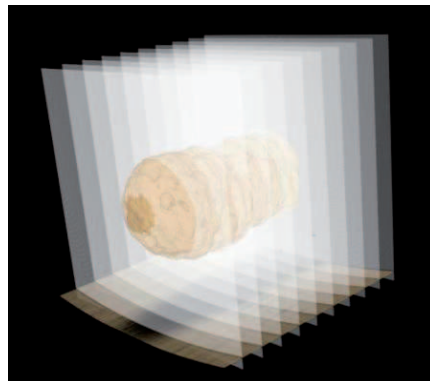


FIG. 2 – Exemple de polygones parallèles alignés. Une texture, correspondant à l'intersection entre le polygone et les données volumiques, est appliquée à chacun d'eux.

En orientant les coupes dans l'axe de vue de la caméra, il n'est pas nécessaire de trier à nouveau les polygones à chaque étape de l'algorithme de visualisation. La rotation autour des données volumiques se réduit donc à l'application d'une rotation aux textures des polygones. Cette opération peut être exécutée de manière très efficace sur les cartes graphiques récentes en utilisant les textures 3D. En effet, une valeur particulière de la texture sur un polygone (dans

Librairie de visualisation

la mémoire de la texture 3D) peut être interpolé directement par la carte graphique à un point sur la surface du polygone le plus proche.

Comme la restitution basée sur l'utilisation d'iso-surfaces, les données volumétriques originales doivent être filtrées pour afficher le volume de l'objet visualisé. Cette opération peut être effectuée de manière interactive sur les cartes graphiques récentes en utilisant les capacités du pipeline programmable (*shader*). L'utilisateur peut ainsi assigner une fonction de transfert pour spécifier une couleur et une transparence à chaque voxel en fonction de son intensité originale.

L'utilisation de cette méthode, au contraire d'approches plus répandues comme le lancer de rayons a l'avantage de permettre l'affichage d'autres rendus en temps réel, ce qui facilite l'ajout d'objets 3D à la visualisation volumique favorisant par exemple l'annotation des données. Il est important de noter que cette méthode peut demander des calculs très intensifs sans l'appui d'une carte graphique récente. En effet la qualité de la visualisation dépend très largement du nombre de coupes utilisées. Si ce nombre est trop petit, des artefacts peuvent apparaître dans la visualisation. Cependant, pour chaque coupe supplémentaire ajoutée, le nombre de calculs augmente de façon considérable (de l'ordre du million pour une image de 512×512 pixels). En revanche, l'optimisation effectuée permet d'atteindre sur du matériel courant aujourd'hui des volumes de $512 \times 512 \times 1024$ voxels codés sur 8 bits, ce qui est sensiblement mieux que sur la plupart des logiciels professionnels médicaux n'autorisant que des volumes de $512 \times 512 \times 512$.

Les capacités de rendu volumique de notre librairie ont également pu être testées dans le cadre du projet MESOMORPH (illustré sur la Figure 3).

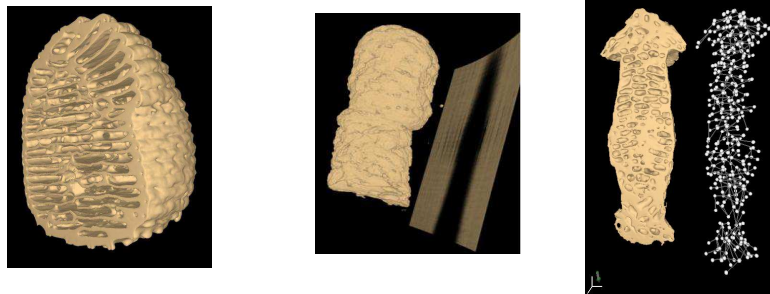


FIG. 3 – Exemple de visualisation de termitières tomographiées. (Gauche) Résultat d'une visualisation volumique indirecte utilisant la méthode iso-surface. (Centre) Résultat d'une visualisation volumique direct utilisant la méthode texture mapping. (Droite) Exemple de visualisation associant différent types de rendus

4 Immersion de l'utilisateur

Quelle que soit l'application, l'interprétation des données visualisées est toujours une tâche délicate. Dans le cas de la visualisation de graphes par exemple, l'occlusion peut impacter l'in-

interprétation car un objet se retrouve caché par un autre. Ce problème peut être contourné en ajoutant des interactions entre l'utilisateur et le système, ou, comme certaines études récentes tendent à le prouver (Ware et Mitchell, 2008; Deligiannidis et al., 2006; Halpin et al., 2008), en immergeant l'utilisateur au sein du système.

Cette section décrit les améliorations introduites dans notre système afin d'améliorer l'expérience utilisateur.

4.1 Visualisation immersive

Une des spécificités de notre framework repose sur l'intégration d'une disparité binoculaire permettant une réelle perception tridimensionnelle des données. En effet, l'homme perçoit le volume et la position relative d'un objet en fonction de différents signaux. Les distances et profondeurs sont ainsi perçues grâce à différents phénomènes physiques incluant la lumière et les ombres, la taille relative des objets, le recouvrement, le gradient de texture, le parallaxe de mouvement, la perspective ou encore la disparité stéréoscopique (Gaggioli et Breining, 2001). Ce dernier signal est considéré comme jouant le rôle le plus important dans la précision de l'estimation des distances relatives à des objets proches.

De ce fait, nous avons intégré plusieurs dispositifs de visualisation stéréoscopique dans notre framework. Celui-ci supporte donc la stéréoscopie passive de type anaglyphe, à visualiser avec des lunettes à filtres ainsi que la stéréoscopie active, à visualiser avec des *shutter glasses* (lunettes alternantes) couplées à un écran ou un projecteur supportant une fréquence d'au moins 100Hz.

Pour immerger d'avantage l'utilisateur, le framework permet également de projeter les images sur un dôme de visualisation. Habituellement, nous utilisons des écrans n'occupant que de 20 à 50 degrés de notre champ de vision. Utiliser un dôme permet d'étendre la surface de projection de façon à ce qu'elle en occupe de 100 à 150 degrés. En plus de l'immersion plus importante, cela permet d'afficher un plus grand nombre d'informations en simultané.

4.2 Interactions

Matériel Les périphériques usuels d'interaction -le clavier et la souris- sont ancrés dans nos habitudes d'utilisation des outils informatiques. Cependant, ils ne sont pas nécessairement adaptés à tout les types d'application. En effet, les applications reposant sur une projection dans un espace euclidien tridimensionnel permettent souvent de se déplacer dans cet espace. Inné dans le monde réel, ce processus peut être délicat dans un monde virtuel car, sans contrainte, il requiert six degrés de liberté là où une souris classique n'en offre que deux.

Il existe différentes techniques permettant d'estomper cette limitation en associant par exemple les mouvements de la souris à une pression de touche quelconque. Celles-ci peuvent se traduire, lorsque la souris est déplacée de la gauche vers la droite, par une translation sur X lorsqu'aucune touche n'est pressée ou par une rotation sur Y lorsque c'est le cas. Toutefois, ceci peut entraîner un déplacement discontinu du fait de l'impossibilité d'effectuer une translation sur X et une rotation sur Y de manière simultanée.

Afin d'essayer de combler ces lacunes, différents périphériques d'interaction ont été développés ces dernières années. Notre système intègre pour l'instant, deux de ces périphériques spécifiques. Le premier, connu sous le nom de souris 3D est un hybride entre un joystick et une souris. Il supporte les six degrés de liberté via l'application d'une pression (de la forme d'une

translation ou d'une rotation) sur la partie joystick du périphérique. Cependant, la position du dispositif est contrainte à une surface plane, ce qui peut être préjudiciable lors d'une visualisation collaborative sur grand écran. Ainsi nous avons également intégré un périphérique issu de l'industrie vidéoludique : la WiiMote de Nintendo. Cette alternative, permet d'offrir à l'utilisateur un dispositif offrant six degrés de liberté grâce aux accéléromètres et gyroscopes intégrés sans avoir à être physiquement situé devant un ordinateur.

Nous offrons également la possibilité de suivre le regard de l'utilisateur de manière stéréoscopique via un procédé d'oculométrie basé sur l'utilisation de deux caméras infrarouges. Ce mécanisme offre ainsi la possibilité d'enregistrer les comportements de l'utilisateur ou encore d'orienter le point de vue de la scène visualisée afin d'améliorer l'impression d'immersion.



FIG. 4 – Exemple d'installation immersive. (Gauche) Dispositif de visualisation immersive dans un dôme. On peut apercevoir au premier plan une souris 3D de la marque 3DConnexion ainsi que deux caméras infrarouge utilisées pour l'oculométrie. (Droite) Visualisation volumique d'un nid de termites avec un système de stéréoscopie actif via un projecteur ACER H3560. Afin de faciliter la présentation, nous pouvons voir qu'une Wiimote est utilisée pour les déplacements dans l'environnement virtuel.

Logiciel Pour faciliter l'adaptation, une librairie de widgets que nous avons développé à été intégré à notre framework. Ces widgets sont des effecteurs représentés dans la projection sous forme d'objets 3D. A l'heure actuelle, les widgets les plus courant sont implémentés : les cases à cocher, les boutons, et les sliders. L'utilité de ces widgets est d'être associé à un ou plusieurs paramètres de la visualisation afin de les faire varier à la guise de l'utilisateur. Par exemple le slider permet de faire évoluer le pas de temps d'une application de visualisation dynamique entre deux valeurs minimum et maximum.

Dans un souci d'optimisation des ressources, les objets 3D modélisant ces widgets sont pour l'instant très simples, mais l'utilisation d'une architecture Modèle-Vue-Contrôleur autorise le développeur à modifier leur apparence sans affecter le reste du framework. De plus, nous pouvons souligner que l'originalité de ces widgets vient de leur nature tri-dimensionnelle. En effet, dans le cadre d'un rendu immersif, ceci permet de ne pas détériorer la qualité de l'expérience immersive. Contrairement aux widgets ordinaires en 2D qui provoquent des artefacts visuels dûs à la différence de profondeur perçue entre les widgets étant au niveau de l'écran et l'objet visualisé qui peut être plus éloigné ou plus proche.

Le matériel immersif supporté est illustré dans la Figure 4.

5 Conclusion

Cet article présente une librairie OpenSource de visualisation interactive et immersive. Cette librairie se focalise sur la facilité d'utilisation et supporte un large éventail de périphériques d'interaction et de méthodes de représentation. Elle intègre des formats très répandus dans différents domaines de recherche, comme les images DICOM, les mesh 3D ou encore les graphes et permet de restituer des données très volumineuses en 3D immersive en temps réel.

Cette librairie est le résultat d'un travail en cours et nous projetons d'y intégrer de nouvelles fonctionnalités. Nous étudions ainsi la prise en charge de CUDA afin d'utiliser les capacités parallèles des cartes graphiques récentes afin de réduire le temps de traitement de données à afficher.

Références

- Auber, D. (2003). Tulip : A huge graph visualisation framework. In P. Mutzel et M. Jünger (Eds.), *Graph Drawing Softwares*, Mathematics and Visualization, pp. 105–126. Springer-Verlag.
Anglais
- Bourke, P. (1999). Calculating stereo pairs. *Published online, <http://astronomy.swin.edu.au/pbourke/stereographics/stereorender/>*.
- Brandes, U. et D. Wagner (2003). Visone - analysis and visualization of social networks. In *Graph Drawing Software*, pp. 321–340. Springer-Verlag.
- Bruneau, P., F. Picarougne, et M. Gelgon (2010). Interactive unsupervised classification and visualization for browsing an image collection. *Pattern Recognition* 43, 485–493.
- Deligiannidis, L., A. P. Sheth, et B. Aleman-meza (2006). Semantic analytics visualization. In *In Proc. of the IEEE International Conference on Intelligence and Security Informatics (ISI-2006)*.
- Dix, A., J. Finlay, G. D. Abowd, et R. Beale (2003). *Human Computer Interaction* (3. ed.). Harlow, England : Pearson.
- Doi, A. et A. Koide (1991). An efficient method of triangulating equivalued surfaces by using tetrahedral cells. *IEICE Transactions Communication, Elec. Info. Syst*, E74(1), 214–224.
- Gaggioli, A. et R. Breining (2001). Perception and cognition in immersive virtual reality. In *In Emerging Communication : Studies on New Technologies and Practices in*, pp. 71–86. IOS Press.
- Greffard, N., F. Picarougne, et P. Kuntz (2010). Immersive dynamic visualization of interactions in a social network. *4th Annual Conference of the German Classification Society (GfKI)*.

- Halpin, H., D. Zielinski, R. Brady, et G. Kelly (2008). Exploring semantic social networks using virtual reality. In *Proceedings of the 7th International Conference on The Semantic Web*, Volume 5318 of *Lecture Notes In Computer Science*, pp. 599–614.
- Kniss, J. M., C. Rezk-salama, D. Weiskopf, et K. Engel (2006). *Real-Time Volume Graphics*. A.K. Peter.
- Lorensen, W. E. et H. E. Cline (1987). Marching cubes : A high resolution 3d surface construction algorithm. *Computer Graphics* 21(4), 163–169.
- McCrickard, D. S. et C. M. Kehoe (1997). Visualizing search results using sqwid. pp. 51–60. ACM Press.
- Misue, K., P. Eades, W. Lai, et K. Sugiyama (1995). Layout adjustment and the mental map. In *J. Visual Languages and Computing*, Volume 6, pp. 183–210.
- Moody, J., D. McFarland, et S. Bender-deMoll (2005). Dynamic network visualization. *American Journal of Sociology* 110, 1206–1241.
- Phong, B. T. (1975). Illumination for computer generated pictures. *Commun. ACM* 18, 311–317.
- Picarougne, F. (2009). Contribution of interactive and immersive visualization techniques for the characterization of major nest designs in social insects. *Proceedings of the International Workshop "From insect nest to human architecture : Engineering principles of innovation in swarm-made architectures"*.
Anglais
- Valverde, S., B. Corominas Murtra, A. Perna, P. Kuntz, G. Theraulaz, et R. V. Solé (2009). Percolation in insect nest networks : Evidence for optimal wiring. *Physical Review E : Statistical, Nonlinear, and Soft Matter Physics*, To be published.
- Ware, C. (2000). *Information visualization : perception for design*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.
- Ware, C. et P. Mitchell (2008). Visualizing graphs in three dimensions. In *ACM Transactions on Applied Perception*, Volume 5, pp. 2–15.

Summary

This paper introduces an Open Source framework of interactive and stereoscopic visualization of complex data. This tool makes the development process of an immersive visualization software easier by handling several interaction peripherals (3D Mouse, wiimote, tracking camera) and several displays (stereoscopic, autostereoscopic and visualization dome). While focusing on the ease of use, this framework acts as a toolbox providing simplified mechanisms allowing the display of 3D data, user interface and interactions. We also present several softwares implemented using this framework.