$N°$d'ordre : 3258

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

par Anthony DON

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Informatique

Indexation et navigation dans les contenus visuels : approches basées sur les graphes

---

Soutenue le : 6 Décembre 2006

Après avis de :

MM.	Yves CHIRICOTA	Professeur	
	Philippe SALEMBIER	Professeur	**Rapporteurs**

Devant la commission d'examen formée de :

MM.	Laurence NIGAY	Professeure	**Présidente**
	Yves CHIRICOTA	Professeur	**Rapporteur**
	Philippe SALEMBIER	Professeur	**Rapporteur**
	Jenny BENOIS-PINEAU ..	Professeure	**Examinatrice**
	Maylis DELEST	Professeure	**Examinatrice**
	Guy MELANÇON	Professeur	**Invité**
	Nicolas HANUSSE	Chargé de Recherche	**Examinateur et rapporteur de soutenance**

– 2006 –

Remerciements

Je remercie les membres du jury, L. Nigay, qui me fait l'honneur de présider le jury de cette thèse, Y. Chiricota et P. Salembier, les rapporteurs de cette thèse, pour leurs remarques qui m'ont permis de considérer mon travail avec plus de recul, et G. Mélançon qui a accepté de participer à ce jury.

Je tiens à remercier particulièrement mes directrices de thèse, Maylis Delest et Jenny Benois-Pineau pour m'avoir fait confiance ainsi que pour le temps et les moyens qu'elles m'ont accordés. Je retiendrais la richesse des enseignements scientifiques qu'elles m'ont transmis ainsi que leurs grandes qualités humaines.

Je remercie du fond du cœur Nicolas Hanusse qui m'a régulièrement aidé, conseillé, encouragé durant ces trois années. Son esprit d'analyse et de synthèse, son imagination, sa curiosité et le temps qu'il m'a consacré ont été des facteurs de succès indiscutables dans l'aboutissement de ce travail.
Merci également aux étudiants de Master qui ont travaillé sur les graphes de navigation, leur travail m'a été très utile : L. Aguerreche, F. Dècle, C. Dongieux, J. Plaisance et S. Roujol.

Ce travail a été réalisé au sein du Laboratoire Bordelais de Recherche en Informatique, dirigé par Serge Dulucq. Je suis reconnaissant des moyens mis à ma disposition ainsi que de l'aide reçue de la part des membres de ce laboratoire, que ce soit dans le cadre des enseignements que j'ai eu le plaisir d'assurer ou pour mes démarches administratives. Je pense en particulier à Giuliana Bianchi, Ida Nerestan, Philippe Biais, Pierre Casteran, Robert Cori, Robert Strandh, ...

Je pense également à D. Auber, J.P. Domenger et N. Novelli pour leurs calembours désopilants et leur bonne humeur. Un salut amical aux anciens membres de la "Salle B" : David Renault (Merci pour ton aide concernant la mise en page de ce document), Marcien Mackaya, Jean-Baptiste Leproux de la Rivière, Bertrand Kerautret, Benjamin Taton. A mes collègues Laura, Sheila, Fanny, Aïda, Maxime, Jocelyn, Jérémie, Olivier, Bilel, Romain, Hayssam, Mamadou, Youssou, Patrick, Mohamed, et tous ceux avec qui j'ai partagé de bons moments.

Un clin d'œil à Nico et Gaëlle, Ben et Marie, Carlos et Marion, Suguna, Jean-Pierre Benedetto, J.F.K., Ringue, Bebbar, Bronsky, Murz et Ricardo.

Une pensée pour ma famille : Hervé et Annick, mon petit frère Alexis (salut mec !), ma sœur Séverine, Sylvie et Abel, Virginie, Ena, Jeanine et Mireille.

Un grand Merci à J.P.M.

Une ligne entière pour Annelise qui fait mon bonheur chaque jour et que j'aime.

Table des matières

Table des figures

Liste des tableaux

Notations

$\|X\|$	Cardinal de l'ensemble X.
D	Ensemble des descripteurs de type D.
(S, δ)	Espace métrique composé de l'ensemble de descripteurs $S \subseteq D$ muni de la mesure de dissimilarité δ.
$B_u(r)$	Boule centrée sur $u \in S$ de rayon $r \in \mathbb{R}^+$.
A	"Aspect ratio" d'un espace métrique.
dd	Dimension doublante d'un espace métrique.
$G = (V, E)$	Graphe composé de l'ensemble de sommets V et de l'ensemble d'arêtes E.
$V(G)$	Ensemble des sommets du graphe G.
$E(G)$	Ensemble d'arêtes du graphe G.
$d_G(u, v)$	Distance dans le graphe G entre les sommets u et v.
T	Arbre d'échantillonage d'un espace métrique.
H_S	Hiérarchie des centres discrets de S.
α	Degré sortant maximum de T.
$C_i(p)$	Ensemble des sommets de S_i ayant p comme parent dans T.
$P(p)$	Parent du sommet p dans T.
$u^{(i)}$	Copie de $u \in S$ au niveau i de la hiérarchie des centres discrets de S.
$P_i(p^{(j)})$	Ancêtre de niveau i du sommet p de niveau j dans T.
S_i	Échantillon des centres discrets de niveau i.
$G_{T,c}$	NAV-GRAPHE construit à partir de l'arbre d'échantillonage T et le coefficient de voisinage c.
$N_{u^{(i)}}(c)$	Voisins du sommet u dans le niveau i de $G_{(S,\delta)}$ situés à distance inférieure ou égale à $c.2^i$.

Acronymes

ACP	Analyse en Composantes Principales
ANMRR	Average Normalized Modified Retrieval Rate
APSP	All-Pair Shortest Path
CBIR	Content-Based Image Retrieval
CCIR	Comité Consultatif International pour la Radio
CLD	Color Layout Descriptor
DAG	Directed Acyclic Graph
DC	Direct Current
DCD	Dominant Color Descriptor
DCT	Discrete Cosine Transform
GEM	Graph EMbedder
GOP/GOF	Group Of Pictures/Group Of Frames
HSV	Hue Saturation Value
ID	Interpretable Descriptor
IHM	Interface Homme-Machine
IRM	Imagerie par Résonance Magnétique
ISO	International Standard Organisation
JND	Just Noticeable Difference
MDS	MultiDimensional Scaling
MLDS	Maximum Likelihood Difference Scaling
MPEG	Moving Picture Experts Group
NIST	National Institute of Standards and Technologies
PCM	Persistence de la Carte Mentale
PHP	PHP Hypertext Preprocessor
QBIC	Query By Image Content
RVB	Rouge Vert Bleu
SFRS	Service du Film de Recherche Scientifique
SQL	Structured Query Langage
SVM	Support Vector Machine
SWIM	Small-World Image Miner
TREC	Text REtrieval Conference
TSL	Teinte Saturation Luminance
TTS	Text-To-Speech
XML	eXtensible Markup Langage

Introduction

Motivation

La grande quantité d'information constituée par les données multimédia numérisées disponibles pose le problème de l'accès rapide aux contenus visuels pertinents dans de grandes collections.

La recherche dans les contenus visuels peut s'effectuer par le contenu textuel ou par le contenu visuel. La recherche par le contenu textuel consiste à utiliser des bases de données pour stocker les images et les informations sémantiques liées à ces images. Le contenu textuel peut être produit manuellement grâce à des outils spécifiques (Flickr [54], PhotoMesa [129]) ou bien être extrait du contexte dans lequel apparaît l'image (Google Image [62]). Les critères utilisés pour la recherche d'un document sont des mots-clé reflétant le concept sémantique que l'utilisateur recherche.

La limite principale de l'annotation est qu'elle reflète seulement un sous-ensemble des concepts sémantiques présents dans une image ou un document vidéo. Par exemple, les caractéristiques précises des objets présents dans une image telles que leur couleur ou la forme des objets ne sont pas décrites lors du processus d'annotation. Pour utiliser ces indices visuels dans le cadre d'une recherche d'image ou de document vidéo, il faut au préalable extraire et résumer ces informations de manière automatique : c'est l'objectif des techniques d'*indexation basées sur le contenu*.

Ce domaine de recherche est très actif et il fait appel à d'autres domaines de l'informatique :

- Analyse d'images : segmentation couleur [35] pour l'extraction de régions, espaces de couleurs et modèles d'apparence couleur [36], extraction et caractérisation des textures [22], des contours, détections de points d'intérêt [112, 101].
- Analyse audio et vidéo : compression et codage, analyse du flot optique[78], modèles de mouvement [48].
- Intelligence artificielle et vision par ordinateur : apprentissage supervisé et non-supervisé pour la détection et la reconnaissance de la parole[128, 138, 148], des caractères, des visages [30] et la classification d'images [159, 102, 25].

Le standard industriel ISO/MPEG-7 [16] est consacré, en partie, au codage et à la composition d'indices visuels pour former des *descripteurs de contenus* afin d'en faciliter la production et l'échange.

La recherche de documents par le contenu visuel s'effectue par similarité visuelle en calculant la distance entre le descripteur visuel (couleur, forme, texture) formant la requête et l'ensemble des descripteurs associés aux documents de la base [22]. Les documents sont ensuite présentés à l'utilisateur par ordre décroissant de similarité avec la requête.

Un problème majeur de ce type de recherche est la difficulté d'exprimer des concepts sémantiques à l'aide de descripteurs visuels exprimant des caractéristiques de bas niveau

issues du document. Le terme de *fossé sémantique* est utilisé pour désigner ce problème et de nombreux travaux de recherche visent à réduire cet écart entre l'utilisateur et les données multimédia numérisées en cherchant des descripteurs de haut niveau sémantique tels que ceux issu de la détection et la reconnaissance de visage [123] ou de la classification des images naturelles en scènes d'extérieur ou d'intérieur [114].

Dans le contexte de l'indexation basée sur le contenu, les documents audiovisuels posent des problèmes spécifiques liés à leur taille importante et à leur structure plus complexe que celle des images fixes. Des standards de codage permettent de réduire leur taille pour en faciliter l'échange et le stockage (standards MPEG-1,2,4) et de nombreux travaux concernent l'analyse automatique de la structure des flux vidéo pour en faciliter l'indexation et la manipulation :

- détection automatique des frontières de plans dans un flux vidéo non-structuré [71, 48],
- caractérisation des mouvements de caméra dans les plans vidéo [48],
- génération automatique de résumés de documents vidéo [158],
- segmentation des objets en mouvement [106],
- détection d'évènements et de scènes spécifiques : séquences de buts dans les programmes sportifs, séquences de reportage dans les journaux télévisés [67], ...

La complexité et la variété des informations qui constituent un document audio-visuel nécessitent donc d'intégrer différents outils d'analyse pour permettre leur indexation.

L'accès à de grandes bases de documents visuels doit reposer sur des structures de données efficaces permettant d'organiser l'ensemble des descripteurs de contenus en vue de leur recherche. Dans ce contexte, les algorithmes d'indexation et de recherche doivent permettre le passage à grande échelle, c'est à dire rester efficaces quand la taille de la collection de documents augmente.

Concernant la création de grands ensembles d'index, les structures de données distribuées utilisées dans les systèmes de stockage et d'échange de fichiers *pair-à-pair* constituent une solution efficace et robuste [10, 117, 133].

Les aspects présentés posent également des problèmes d'interaction et de visualisation d'information dans les scénarios suivants :

- fragmentation de graphes (cf. Définition 10 pour identifier de manière automatique des fragments [149, 121] ou des communautés d'objets similaires[168, 7],
- plongement de graph (cf. Définition 40) pour refléter fidèlement la structure des collections d'objets manipulées [151, 60],
- manipulation et affichage de grandes collections de contenus visuels [129, 110],
- formulation de requêtes pour la recherche par similarité visuelle [53, 26, 98, 32],
- accès non-linéaire aux documents vidéo [158, 94, 77].

Le thème de l'accès aux contenus multimédia indexés est donc un thème transversal qui concerne différents domaines de recherche.

But de notre travail

Dans cette thèse, nous abordons chacun des aspects de la recherche d'information visuelle [22]. Nous nous intéressons à l'extraction d'indices visuels et donc au domaine de l'*indexation* basée sur le contenu. Dans ce contexte, nous examinons aussi l'analyse de la structure des documents vidéo et leur fragmentation. Nous traitons également la visualisation des collections d'images indexées ainsi que les interfaces de recherche dans ces collections. Enfin, nous proposons des outils de test pour la validation des méthodes

proposées. Notre vision de ces problèmes se basera principalement sur une modélisation par des graphes.

Dans la suite de cette thèse, nous considérons que pour un descripteur de type D (couleur, texture, forme ...), nous disposons d'une métrique δ telle que pour toute paire de documents x, y, nous sommes capables de calculer $\delta(x, y)$, la mesure de dissimilarité entre les documents x et y calculée en fonction des indices visuels qui composent le descripteur D. On manipule l'espace métrique (S, δ), avec $S \subset D$, pour agir sur les descripteurs d'une collection de documents indexés (images ou plans vidéo).

Concernant l'analyse de la structure d'une séquences vidéo brute, l'objectif est de produire une segmentation d'un document vidéo en groupes de plans vidéo partageant une sémantique commune. Cette segmentation est ensuite utilisée pour proposer une visualisation et un accès non-linéaire au document vidéo. On parlera de *scènes vidéo* [85] lorsque les plans groupés sont consécutifs et d'*hyper-scènes* quand ces plans sont non-contigus dans le document vidéo.

Dans notre contexte, l'espace métrique composé de l'ensemble des plans vidéo indexés est modélisé par un graphe complet valué que nous appelons *graphe vidéo*. L'identification des hyper-scènes est interprétée comme un processus de filtrage des arêtes de faible similarité du graphe vidéo suivi d'une recherche de sous-graphes denses dans le graphe filtré. Cette approche s'apparente à la fragmentation "naturelle" de graphe qui consiste à identifier de manière automatique des fragments correspondant à des groupes d'objets similaires [149, 121, 168, 7] .

Nous abordons également la détection de *scènes de dialogues* dans les documents vidéo. Pour cela, nous considérons la recherche de motifs périodiques dans la matrice d'adjacence du graphe vidéo. Cette méthode tire parti d'une indexation de haut niveau constituée par la détection des visages dans les plans vidéo.

Concernant la visualisation et la recherche de documents indexés, nous nous basons sur un *arbre d'échantillonage* de l'espace métrique (S, δ). Cette structure de données est exploitée par un algorithme de placement utilisé pour visualiser les groupes de documents similaires. Pour comparer notre placement avec d'autres algorithmes de la littérature, nous proposons un algorithme de génération de jeux de données de test basé sur la génération aléatoire d'arbres.

Nous exploitons également l'*arbre d'échantillonage* dans la construction d'un *graphe de navigation* qui constitue une structure de graphe adaptée pour la recherche visuelle d'un élément dans (S, δ). Nous proposons une interface de *navigation locale* dans une collection d'images indexées ainsi qu'un cadre expérimental pour valider cette approche.

Indexation de documents vidéo

La production d'œuvres télévisées ou cinématographiques obéit à des règles de production. Le terme de *grammaire cinématographique* [4] est utilisé pour désigner l'ensemble des conventions utilisées pour la réalisation d'un film. Cependant, les règles de cette grammaire sont générales et laissent une place importante à la créativité du réalisateur. Les mêmes règles de production peuvent donc conduire à des contenus audio-visuels dont les caractéristiques seront très différentes. L'analyse des flux audio-visuels dans le but d'en reconstruire la structure doit tenir compte de la diversité de forme qui peut exister dans les documents issus de la même grammaire.

Pour analyser les scènes présentes dans un document audiovisuel, des méthodes contraignent le problème en supposant l'existence d'un modèle de document particulier [67, 150]. Les méthodes basée sur un modèle sont conçues pour extraire des scènes sémantiques is-

sues de documents vidéo d'un genre spécifique : séquence de reportage dans les journaux télévisés [67], séquences de buts dans un match de football, scène composée d'évènements en série ou en parallèle dans les longs métrages [150].

Les méthodes génériques ne se basent pas sur une hypothèse quant au genre du document vidéo analysé. Elles se basent sur la détection de motifs particuliers dans l'enchaînement des plans vidéo [148] ou sur la détection et la reconnaissance de visages [128] pour détecter des scènes de dialogue. La fusion d'informations issues de l'analyse des flux audio, vidéo et des sous-titres est aussi utilisée pour délimiter les scènes d'un document vidéo [1, 75].

Dans ce contexte, nous nous intéressons à la modélisation des relations de similarité des plans d'un document vidéo par un graphe. Notre approche consiste ensuite à analyser la structure du graphe pour extraire des informations utiles du point de vue du document vidéo. Nous avons notamment effectué le groupement des plans d'un document vidéo en scènes et hyper-scènes homogènes en termes de couleur.

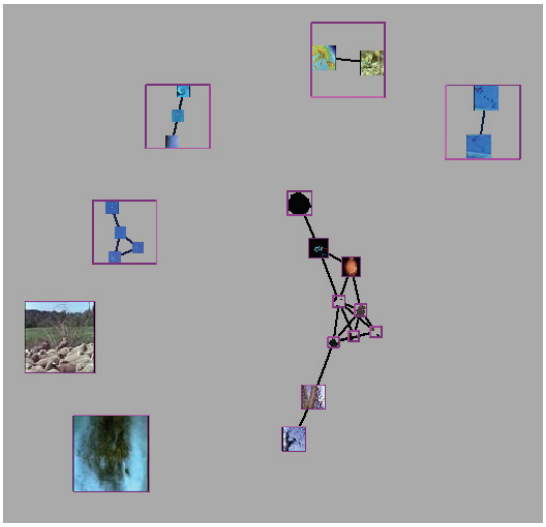


FIG. 0.1: Fragments d'un graphe vidéo.

Algorithme de placement

L'identification des groupes de documents similaires dans une collection permet d'offrir une visualisation proposant d'abord une vue globale sur les groupes puis permettant à l'utilisateur de filtrer les groupes non-pertinents. Cette approche suit un concept-clé de la visualisation d'information énoncé par B. Shneiderman : "Overview, zoom and filter" [143].

Pour proposer ce type de visualisation, un plongement en dimension 2 ou 3, qui reflète la similarité entre les documents indexés, peut être utilisé pour organiser une collection d'images indexées de manière automatique. La principale difficulté consiste à trouver un plongement qui engendre la plus petite distorsion possible entre les distances dans l' espace (S, δ) et les distances correspondantes dans le plongement.

Le "Multidimensional Scaling" (MDS) est un ensemble de techniques issues des statistiques et utilisées en visualisation d'informations pour explorer les similarités et les dissimilarités d'un jeu de données. Les algorithmes de MDS [60, 69, 58, 50, 119, 31, 86, 57]

représentent une solution au problème de la représentation des relations de similarité dans une collection de documents indexés.

Notre objectif est de proposer une visualisation d'une collection de documents indexés avec une faible distorsion entre la distance dans l'espace métrique et la distance euclidienne dans le plongement de cet espace en deux dimensions. Pour que cette technique soit adaptée à la représentation d'entités visuelles telles que des images, nous avons développé un algorithme qui ne permet pas les chevauchements entre ces entités, dans le plongement.

Recherche exploratoire et graphe de navigation

La recherche d'informations visuelles dans les grandes collections d'images ou de documents multimédia indexés nécessite des techniques spécifiques pour la formulation et le raffinement des requêtes. Des techniques de *requêtes par l'exemple* consistent à spécifier la valeur du descripteur donnant les caractéristiques du document recherché :

- en donnant une instance de document similaire au document recherché duquel un descripteur est extrait ("query-by-example") [53, 26],
- en fixant les valeurs numériques du descripteur ("query-by-feature") [53],
- en effectuant un croquis du document recherché à partir duquel le descripteur sera extrait ("query-by-sketch") [32, 98].

La principale limitation des techniques de requêtes par l'exemple réside dans le fait que l'utilisateur doive fournir un exemple souvent trop précis du document recherché.

Les techniques de bouclage de pertinence ("relevance feedback") forment une autre catégorie de techniques de recherches visuelles [26, 110]. Ces techniques consistent à demander à l'utilisateur de désigner un ensemble de documents pertinents et non-pertinents dans un échantillon de la collection de documents. Un score de pertinence basé sur ces exemples est ensuite calculé et les documents ayant le meilleur score sont proposés à l'utilisateur. Cette procédure est itérée jusqu'à ce que le document recherché soit trouvé. La principale limitation du bouclage de pertinence est liée à la difficulté de trouver des exemples positifs quand l'ensemble d'images constituant la cible recherchée est de petite taille en comparaison avec la collection complète.

Pour les raisons évoquées, les techniques permettant une *recherche exploratoire* sont utiles lorsque la requête ne peut pas être formulée avec précision. Nous proposons d'envisager la recherche d'une image comme un processus de navigation dans un graphe qui modélise une collection d'images : les sommets sont les images de la collection et des arêtes "utiles" à la navigation sont ajoutées. La notion d'utilité des arêtes est liée au type de navigation dans le graphe.

Nous souhaitons que le graphe de navigation permette d'effectuer une navigation locale s'apparentant au "routage glouton" dans les réseaux où une succession de décisions basées sur une information locale à chaque nœud du réseau permet d'acheminer un paquet d'information vers un nœud précis du réseau. Dans notre contexte, le réseau de machines est remplacé par un réseau d'images indexées dont la topologie sera induite par les valeurs des descripteurs utilisés.

La figure 0.2 illustre le processus de navigation locale. La destination est constituée par l'image recherchée (sommet rouge) et l'utilisateur joue le rôle de l'algorithme de routage et choisit, à chaque étape, le sommet voisin (sommets oranges) du sommet courant (sommet vert) qui partage le plus de caractéristiques avec le sommet recherché.

La structure de graphe utilisée, le schéma d'indexation et la visualisation de chaque vue locale sont les caractéristiques-clé d'un tel système de navigation.

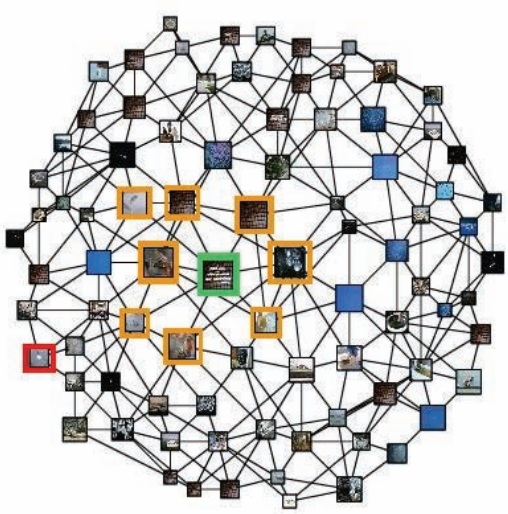


FIG. 0.2: Illustration du graphe de navigation.

Indices visuels interprétables

Un programme qui implémente un algorithme de *routage glouton* et un humain appliquant ce routage dans le contexte de la recherche visuelle diffèrent dans le sens où le programme déterminera toujours le sommet voisin le rapprochant de sa destination (en calculant les valeurs de dissimilarité). A cause des caractéristiques de la perception visuelle et du "fossé sémantique", l'humain commet des erreurs et ne suit pas toujours le plus court chemin vers sa destination. Pour faciliter la tâche de l'utilisateur, la définition d'un descripteur spécifique est nécessaire.

Les indices visuels utilisés pour former le descripteur doivent permettre à l'utilisateur d'évaluer la valeur du descripteur en examinant le contenu de l'image. Nous définissons donc des *indices visuels interprétables* pour former les descripteurs utilisés pour construire les graphes de navigation.

Paramètres intrinsèques des données

Des travaux récents sur des structures de données efficaces [93, 21] pour la recherche des points les plus proches d'une cible dans un espace métrique tiennent compte de la *dimension doublante* de l'espace métrique (cf. Définition 47) pour le calcul des complexités des algorithmes de recherche.

La dimension doublante de (S, δ) caractérise la dimension intrinsèque des données associées à l'espace métrique. Cette notion de dimension diffère de la dimension apparente des données. En effet, un descripteur peut être composé d'un grand nombre d'attributs numériques, mais sa dimension intrinsèque, caractérisée par sa dimension doublante peut s'avérer très faible.

Dans cette thèse, nous ferons souvent référence à la *dimension doublante*, dd de l'espace métrique (S, δ) car cette notion influence également les performances et les temps d'exécution des algorithmes que nous proposons.

L"'aspect ratio", A, de l'espace métrique représente le rapport entre distance maximum et distance minimum dans (S, δ) (cf. Définition 43). Ce paramètre influence également les complexités de nos algorithmes.

Contributions

Fragmentation de graphe

Nous présentons une méthode d'analyse des hyper-scènes couleur d'un document vidéo dans le chapitre 2. Nous utilisons la *signature couleur spatio-temporelle* des plans vidéo pour caractériser le contenu couleur de l'ensemble d'un plan vidéo [42]. Nous proposons ensuite une méthode de groupement des plans en scènes basée sur la fragmentation de graphe pour identifier des hyper-scènes colorées dans le document vidéo.

Les résultats obtenus sur quatre films documentaires sont comparés avec une indexation manuelle de référence des plans vidéo en hyper-scènes couleur. Ces résultats indiquent que la méthode est adaptée à l'analyse de documents vidéo contenant des scènes couleur fortement contrastées.

Indexation des scènes de dialogue

Dans le chapitre 3, nous proposons une méthode originale de détection de scènes de dialogue basée sur la fusion de deux informations issues du flux vidéo :

– l'analyse des motifs périodiques persistants dans l'enchaînement des plans vidéo,
– la détection de visages dans les plans vidéo.

Concernant la détection de visages, notre approche met en œuvre une collaboration entre deux détecteurs de visages. L'un est basé sur un algorithme de classification supervisée, les machines à vecteur de support (SVM), l'autre, utilise la modélisation de la couleur de peau dans un espace de couleur approprié (YCbCr). Le paramétrage du modèle de couleur de peau utilise les résultats du premier détecteur, ce qui permet d'adapter le modèle de couleur aux caractéristiques spécifiques du document vidéo analysé et d'améliorer la précision des résultats produits par le premier détecteur.

Algorithme de placement

Le chapitre 4 présente un algorithme de plongement d'un espace métrique en deux dimensions appliqué à la visualisation d'images indexées. La complexité théorique de notre méthode est quasi-linéaire et permet donc un affichage rapide. Nous montrons également en quoi la dimension doublante et l"'aspect ratio" du jeu de données utilisé constituent des "paramètres cachés" qui influencent le résultat de notre algorithme. La complexité totale de l'algorithme de plongement d'un espace métrique avec n éléments et d'"aspect ratio" A est en $O(n \log A)$

Comme l'ajout de nouveaux éléments dans l'arbre n'altère pas le plongement existant dans sa totalité, la représentation associée peut être mémorisée et servir de support à la carte mentale de l'utilisateur pour l'accès au contenu de la collection.

Navigation

La méthode de recherche visuelle dans une collection d'images indexées basée sur le paradigme de navigation locale dans un graphe est présentée dans le chapitre 5. La struc-

ture du *graphe de navigation* utilisée est calculée à partir de l'ensemble des descripteurs associés aux images de la collection indexée.

L'analyse théorique concernant le temps de recherche d'une cible dans ce graphe de navigation indique un temps de recherche influencé par :

- $\log_2 A$, le logarithme de l'"aspect ratio" de l'espace métrique, qui borne le nombre de sauts à effectuer dans le meilleur des cas,
- le nombre d'images semblables à la cible recherchée, qui peut contraindre l'utilisateur à parcourir un ensemble d'images considérées comme indistinguables au sens du descripteur utilisé avant de trouver la cible.

Les expériences de recherche d'images dans une collection d'images indexées que nous avons conduites avec des utilisateurs-test montrent l'efficacité de notre approche par rapport à une approche de recherche naïve de complexité linéaire.

Au début de cette thèse, nous envisagions d'utiliser les descripteurs visuels ISO/MPEG-7 dans le cadre de la navigation. Au cours de nos travaux, ces descripteurs sont apparus comme difficilement interprétables par un humain : un utilisateur pouvant difficilement choisir l'image la plus proche d'une image cible en interprétant ce type de descripteur (histogramme couleur ou coefficients fréquentiels). Pour résoudre ce problème, nous proposons un schéma d'indexation adapté au contexte de la navigation locale. Nous proposons un descripteur d'images composé de la concaténation de six caractéristiques choisies pour leur facilité d'interprétation : nombre de visages dans l'image, surface occupée par les visages de l'image, nombre de régions dans l'image, luminance moyenne de l'image, valeur de teinte et de saturation de la couleur la plus saturée.

Ce descripteur interprétable permet d'évaluer la valeur du descripteur en visualisant l'image associés, ce qui facilite la navigation locale. Les résultats expérimentaux montrent que l'efficacité perçue par les utilisateurs ainsi que l'efficacité mesurée donne l'avantage à ce descripteur plutôt qu'au descripteur ISO/MPEG-7 "Color Layout Descriptor" dans le contexte de la navigation.

Une caractérisation précise de la manière dont ces caractéristiques sont perçues par l'utilisateur permet de pondérer la contribution de chaque indice visuel dans le descripteur composite. Nous proposons une estimation de la perception de chaque indice visuel selon une méthodologie issue de la psychologie expérimentale (cf. annexe A).

Génération aléatoire de jeux de données de test

Dans les chapitres 4 et 5, nous montrons l'impact de la *dimension doublante* d'un espace métrique sur les performances des algorithmes de plongement de métriques et sur la recherche dans les graphes de navigation dans ces espaces métriques.

La comparaison à grande échelle, des différentes méthodes, expériences et algorithmes présentés dans cette thèse nécessite plusieurs jeux de données ayant la même dimension doublante. La technique de comparaison envisagée est la simulation sur des jeux de données générés aléatoirement. Cependant, nous sommes confrontés à la difficulté de générer des espaces métriques à dimension doublante fixée. Dans la section B, nous montrons des relations entre le degré maximum d'un arbre construit à partir d'un espace métrique et la dimension doublante de l'espace métrique. En l'occurrence, une petite dimension doublante implique un petit degré maximum.

Dans l'annexe C, nous montrons comment générer, *en temps linéaire*, un grand nombre d'espaces métriques de dimension doublante donnée. Notre technique se base sur la génération d'un arbre à partir duquel un espace de dimension doublante donnée est construit.

La distribution des degrés de l'arbre permet de paramétrer la valeur de la dimension doublante de l'espace métrique généré. Cette dernière propriété est importante car elle permet de reproduire fidèlement des espaces métriques observés dans nos jeux de données réels. Nous utilisons les espaces métriques générés pour évaluer l'algorithme de placement décrit dans le chapitre 4.

Chapitre 1
Présentation des domaines

Dans ce chapitre, nous présentons les domaines abordés dans cette thèse, les notions et les notations essentielles à la compréhension des chapitres suivants. La première section présente un panorama des concepts et des techniques employés dans le domaine de l'indexation des contenus multimédia qui est le thème principal de cette thèse. La deuxième section introduit quelques notions et des notations sur les graphes utilisées dans les chapitres suivants. Dans la section suivante, nous introduisons les concepts et les travaux associés à la visualisation d'information qui est un domaine dans lequel s'inscrivent certaines de nos contributions. Dans l'avant-dernière section, nous détaillons différents algorithmes de "multidimensionnal scaling", une technique de visualisation par plongement de données en deux dimensions. La dernière section est consacrée à l'explication d'une technique d'échantillonage d'un espace métrique utilisée dans les chapitres 4, et 5 ainsi que dans l'annexe C.

1.1 Indexation et recherche d'information basées sur le contenu

La recherche d'images indexées basée sur le contenu ("Content-based image retrieval" (CBIR)) désigne l'ensemble des techniques permettant la recherche d'images et de documents audiovisuels dans de grandes bases de données. Le terme "indexation basée sur le contenu" signifie que le processus de recherche utilise des informations extraites du contenu des images elles-mêmes plutôt que des meta-données (mot-clés, descriptions) ajoutées manuellement.

Un système CBIR idéal, du point de vue de l'utilisateur, devrait permettre d'effectuer des requêtes sémantiques telles que "trouve des images contenant une voiture rouge". Mais ce type de requêtes est très difficile à mettre en œuvre à cause de la diversité des instances de chaque concept dans les images. Les systèmes actuels reposent essentiellement sur l'analyse des caractéristiques de bas-niveau issues de l'image telles que les couleurs, les textures et les formes. On appelle "fossé sémantique" ("semantic gap" [66]), la difficulté d'exprimer des concepts sémantiques à l'aide de caractéristiques de bas niveau. Des caractéristiques de plus haut-niveau telles que la présence de visages dans l'image ou la reconnaissance de l'orientation de l'image (portrait/paysage) sont également utilisées dans les systèmes CBIR pour enrichir l'indexation.

En plus des techniques d'analyse et d'indexation, de nouveaux concepts d'interaction sont requis pour permettre l'accès à des contenus visuels [22]. Les requêtes des utilisateurs peuvent prendre différentes formes :

– requêtes par l'exemple ("query by example"), pour lesquelles l'utilisateur fournit une image représentative des images recherchées. Cette image peut être produite par l'utilisateur ou issue d'un échantillonage de la base d'images. Le système recherche ensuite les images de la collection les plus similaires à la requête selon les caractéristiques indexées [26, 98].
– requêtes par croquis, ("query by sketch"), pour lesquelles l'utilisateur crée un dessin du document recherché. L'interface utilisateur doit alors permettre de dessiner les zones qui composent l'image et de fixer leurs attributs de couleur, texture ou trajectoire [32, 53, 98].
– D'autres méthodes permettent de quantifier directement les valeurs numériques associées aux caractéristiques de l'image recherchée ("query by features"). Par exemple, son histogramme de couleur [53].

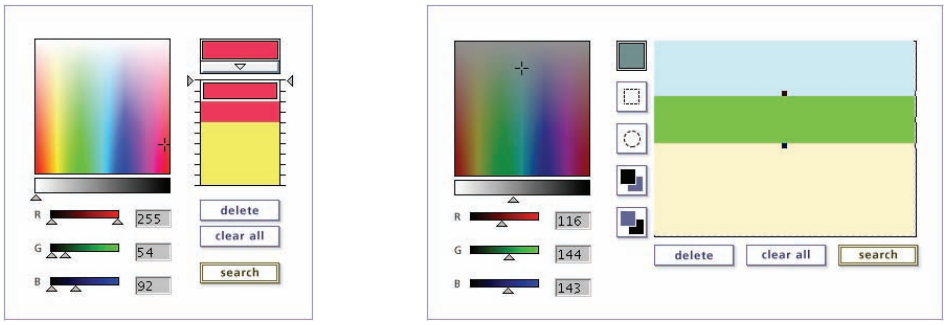
Des techniques d'interaction sont également utilisées pour aider l'utilisateur à formuler et reformuler ses requêtes. Les techniques de bouclage de pertinence ("relevance feedback") impliquent l'utilisateur dans un cycle de raffinements de sa requête initiale. A partir d'un ensemble d'éléments qui lui sont présentés, l'utilisateur indique la pertinence de chaque élément. La requête est alors adaptée pour tenir compte des informations fournies par l'utilisateur [26, 110].

Nous proposons maintenant une revue des travaux et applications développées dans le contexte de l'indexation basée sur le contenu.

1.1.1 Collections d'images

Les interfaces de recherche dans les collections d'images basées sur une indexation par le contenu concernent essentiellement des applications professionnelles.

Les peintures du musée de l'Hermitage à Saint-Pétersbourg [53], sont indexées grâce au système QBIC [81]. Les requêtes peuvent être formulées grâce aux caractéristiques couleur des tableaux (figure 1.1.a) ou au travers d'une interface permettant de réaliser un croquis de l'œuvre recherchée (figure 1.1.b).



a) b)

FIG. 1.1: Interface pour la formulation de requêtes dans QBIC. a) Spécification de la requête à l'aide de l'histogramme couleur. b) Spécification de la requête à l'aide d'un croquis couleur.

De nombreux travaux concernent l'indexation et l'interrogation de collections d'images médicales. La recherche de caractéristiques discriminantes adaptées à la nature particulière des clichés médicaux est plus particulièrement traitée dans ce contexte (Segmentation et indexation de régions d'intérêt [140, 173]. Détection et indexation de points d'intérêt [113]).

Dans le contexte de la protection de la propriété industrielle, des outils basés sur l'indexation par le contenu ont été proposées. Dans [80], les auteurs proposent d'indexer les schémas techniques des brevets à l'aide du graphe composé des lignes du schéma. L'indexation et la recherche de logo et de marques de fabrique dans des images est abordée dans [172].

La plupart des outils efficaces dédiés à la recherche dans les collections d'images concerne essentiellement des domaines industriels et des problèmes très contraints. Dans le contexte plus général de la recherche d'images naturelles ou de photographies personnelles, la contribution des techniques CBIR est très limitée. Cela est essentiellement dû au fossé sémantique qui empêche d'exprimer des requêtes de haut niveau sémantique à partir des caractéristiques de bas niveau des images.

Des techniques de classification sont utilisées pour produire des informations de plus haut niveau sémantique à partir du contenu des images. La détection de visages basée sur les machines à vecteur de support (SVM) est l'exemple le plus représentatif de cette approche [126, 30]. Des systèmes de classification permettent de distinguer l'orientation des images (portrait/paysage) [159], l'environnement du contenu de l'image [102](intérieur/extérieur), ou la nature du contenu [25](naturel/artificiel).

Dans [114], un ensemble de catégories sémantiques est déterminé à partir d'un ensemble de descripteurs de bas niveau. Cet ensemble comporte 40 caractéristiques issues de l'analyse de l'image (nombre de régions après segmentation, fréquences spatiales, présence de lignes droites, etc.) et permet de définir 20 catégories sémantiques (feuillage, foules, paysages avec présence d'eau, paysages urbains, etc.).

Les produits destinés au grand public ont dans un premier temps utilisé le système de fichiers pour l'organisation des images et des autres types de fichiers. Des visionneuses d'images ont donc été développées pour permettre l'examen visuel des images présentes dans une hiérarchie de répertoires. La prise en compte du plus grand nombre de formats d'images est une fonctionnalité centrale pour la construction des icônes associés aux images. Des produits comme ACDSee, Picasa et Explorer offrent ces fonctionnalités.

Une évolution majeure apportée au mode de navigation dans une application commerciale est proposée dans PhotoMesa [14]. Cette interface organise des groupes d'images issus d'un système de fichiers en une vue en deux dimensions d'après un algorithme de "Quantum Treemap", une évolution des "Treemaps" [14] qui produit des zones rectangulaires contenant une vue iconique d'une collection d'images avec un "aspect-ratio" proche de un. L'utilisateur peut ainsi obtenir une vue globale de sa collection d'images dans une vue unique (cf. figure 1.2).

Les propositions récentes concernant l'organisation et l'exploration des collections de photographies personnelles se basent essentiellement sur les annotations automatiques et collaboratives des collections d'images.

FlickR [54] permet de mettre en ligne des photographies personnelles et de permettre à un groupe d'utilisateurs d'annoter certaines zones de l'image et d'y associer des mots-clé prédéfinis (tags). La recherche d'une image se fait par mots-clé. Dans WWMX [111], les utilisateurs peuvent partager des photographies en les localisant géographiquement. L'interface utilisateur permet d'accéder aux photographies d'un lieu en particulier.

Une autre tendance concerne l'utilisation de facettes ou de catégories indépendantes prédéfinies pour faciliter l'indexation des collections d'images [76].

Une initiative intéressante pour l'annotation d'images se présente sous la forme d'un jeu en ligne ("the ESP game") où deux participants anonymes annotent la même photographie. Les joueurs marquent des points lorsqu'ils proposent le même mot-clé [163]. Dans [63], les

photographies sont organisées et présentées sous forme de fragments associés à leur date de création.

FIG. 1.2: Interface utilisateur du logiciel PhotoMesa.

1.1.2 Documents audiovisuels

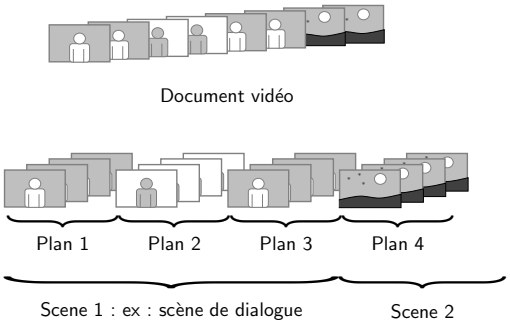


FIG. 1.3: Modèle de document vidéo.

Un flux vidéo est constitué d'une succession d'images, échantillonnées toute les 1/25 ou 1/30 seconde. L'image constitue l'unité d'information vidéo. Selon les règles de production audiovisuelles [85], un document vidéo peut être décomposé en une hiérarchie de scènes et de plans. Un plan correspond à l'ensemble d'images situées entre le démarrage et la coupure de la caméra, ou entre des effets de transition. Les plans sont les composants élémentaires de la vidéo. Cependant, un plan vidéo peut être si long et complexe que sa subdivision en séquences plus petites peut être nécessaire. Un plan peut donc être subdivisé en micro-plans, correspondant à une succession d'images caractérisées par un mouvement de caméra homogène [85].

Des plans consécutifs et dont le contenu partage les mêmes caractéristiques de lieu, de temps ou d'action peuvent être groupés en scènes [85]. Le problème complexe de la définition de scène dépend fortement du type de document concerné [176, 150].

La figure 1.3, illustre la décomposition d'un document vidéo en plans et scènes.

Dans la pratique, les méthodes automatiques de groupement des plans en scènes supposent que la sémantique visuelle peut être suffisamment bien représentée par des descripteurs de bas niveau. Ainsi, une distribution spécifique de couleur dans le plan image

permet sûrement de séparer scènes d'intérieur et vues extérieures, une bonne mesure de l'activité de mouvement permet de distinguer les scènes d'action et des contenus narratifs presque statiques etc.

Les contenus audiovisuels se distinguent notamment des collections d'images par l'ajout de la dimension temporelle et d'un flux audio. Cette complexité supplémentaire pose des problèmes nouveaux concernant l'indexation, la recherche et l'accès aux documents vidéo.

Le standard ISO/MPEG-7 [16] vise à faciliter l'utilisation des descriptions issues des contenus multimédia en normalisant le codage de leurs caractéristiques ainsi que leur composition.

L'accès à un document vidéo repose essentiellement sur la structuration automatique du document vidéo. Les techniques utilisées consistent à détecter les frontières des plans vidéo et les frontières des scènes. Des informations supplémentaires peuvent alors être associées à ces segments (descripteurs de couleur, de mouvement, mot-clés). Un degré de raffinement supplémentaire consiste à analyser la présence d'objets dans les plans vidéo, par exemple un personnage, un véhicule, ou plus généralement une zone de l'image dont le mouvement diffère du mouvement global du plan vidéo [106]. Les objets peuvent alors être caractérisés par des descripteurs de trajectoire ou de couleur et être indexés indépendamment.

1.1.3 Caractérisation spatio-temporelle des plans vidéo

Les caractéristiques d'un plan vidéo peuvent être décrites de différentes manières. Une ou plusieurs images représentatives (ou images-clé) peuvent être extraites de l'ensemble des images composant le plan vidéo et l'ensemble des descripteurs d'images associés peut être utilisé pour décrire le plan.

L'union des caractéristiques issues de chaque "frame" peut être employé pour caractériser globalement un plan vidéo. Le standard ISO/MPEG-7 définit à cet effet le "Group Of Frames Descriptor" qui correspond à l'histogramme des couleur issues d'un ensemble d'images [15]. Les mosaïques spatio-temporelles sont également utilisées pour caractériser globalement un plan vidéo [171].

L'analyse du mouvement dans le plan vidéo représente une information importante pour l'indexation des plans vidéo. L'analyse du flot optique est fréquemment utilisée pour déterminer la valeur de ces descripteurs. Le flot optique consiste à calculer un vecteur de vitesse pour chaque pixel [78] ou blocs de pixels d'une "frame". Cela donne lieu à un champ dense de vecteurs à partir duquel on peut extraire les caractéristiques du mouvement 3-D [48].

Le standard ISO/MPEG-7 définit différents descripteurs de mouvement pour caractériser le mouvement global dans le plan vidéo ("Motion Activity Descriptor"), le mouvement de caméra ("Camera Motion Descriptor" et "Parametric Motion") et la trajectoire des objets dans l'image ("Motion Trajectory") [17].

La caractérisation spatio-temporelle d'un plan vidéo consiste à tenir compte de l'évolution d'une caractéristique du plan vidéo au cours du temps [154, 85].

Ainsi, la notion de signature spatio-temporelle d'un plan vidéo peut être introduite. Parmi les différentes définitions possibles, nous allons présenter la *signature couleur spatio-temporelle* d'un plan vidéo qui est un descripteur spécifique des plans vidéo qui permet d'en mesurer la similarité couleur. Cette signature [19] est basée sur une projection discrète en une dimension appelée "Projection X-Ray". L'image "X-Ray" d'une "frame" vidéo a été introduite par Tonomura [154].

Définition 1 (Image couleur discrète)
Soit une zone rectangulaire de largeur W et de hauteur H, et soit $\mathbb{N}^3$ un espace couleur discret à 3 composantes. Une image couleur discrète, notée $(I(x,y))_{1\leq x\leq W, 1\leq y\leq H}$, est une matrice dont les éléments sont : $\forall x \in [0,W], y \in [0,H], I(x,y) \in \mathbb{N}^3$.

Définition 2 (Projection "X-Ray")
Soit $(I(x,y))_{1\leq x\leq W, 1\leq y\leq H}$ un image couleur discrète de W pixels de large et H pixels de haut. La valeur moyenne des pixels de I selon la ligne d'indice y, resp. selon la colonne d'indice x, est notée $\mathcal{X}_x^I(y)$, resp. $\mathcal{X}_y^I(x)$, et définie par :

$$\mathcal{X}_x^I(y) = \frac{1}{W}\sum_{k=1}^{W} I(k,y) \tag{1.1}$$

resp.,

$$\mathcal{X}_y^I(x) = \frac{1}{H}\sum_{k=1}^{H} I(x,k) \tag{1.2}$$

On appelle les valeurs $\mathcal{X}_x^I(y)$ et $\mathcal{X}_y^I(x)$ des "bins". Les ensembles de "bins" définis par $\{\mathcal{X}_x^I(y)|\forall y \in [1,H]\}$ et $\{\mathcal{X}_y^I(x)|\forall x \in [1,W]\}$ sont des projections "X-Ray" de I selon l'axe vertical, resp. horizontal.

Dans le cas des images en couleur, le "bin" résultant de la projection de la ligne d'indice y ligne de la "frame" I est un vecteur à trois composantes $\mathcal{X}_x^I(y) = (c_1, c_2, c_3)$ dans l'espace couleur discret (C_1, C_2, C_3). Les valeurs d'un "bin" en couleur sont obtenues par projection indépendante des trois composantes couleur de la "frame" couleur. L'espace couleur utilisé ici est l'espace YUV et seule la direction de projection horizontale est utilisée pour obtenir la transformée "X-Ray" d'une image. Ainsi, l'application de la formule 1.2 sur une "frame" couleur $(I(x,y))_{1\leq x\leq W, 1\leq y\leq H}$ réduit la "frame" à une colonne de H "bin" que l'on peut écrire sous la forme d'un vecteur :

$$\mathcal{X}_x^I = (\mathcal{X}_x^I(1), ..., \mathcal{X}_x^I(H))^T \tag{1.3}$$

Souvent, on considère seulement la projection selon la direction horizontale car cette direction permet de refléter au mieux la structure naturelle des scènes les plus courantes. Ainsi, un paysage de campagne verdoyante sous un ciel bleu produira un vecteur contenant des "bins" bleu en haut et des "bins" verts en bas. Dans [122], les auteurs caractérisent les plans vidéo par une mosaïque de descripteurs de type "X-Ray".

La plupart des contenus vidéo étant disponible en format compressé, il est possible de limiter les traitements effectués sur les images en résolution DC (Direct Current) qui peuvent facilement être extraites des flux compressés MPEG sans les décoder complètement. Une image en résolution DC est une version de l'image en pleine résolution dont la taille est réduite 8 fois dans chaque dimension. La valeur d'un pixel d'une image DC correspond à la moyenne d'un bloc de 8×8 pixels de l'image en pleine résolution.

1.1.3.1 Indexation vidéo automatique

L'indexation vidéo automatique est un domaine de recherche très actif. Un projet précurseur dans ce domain est le projet Informedia [75] qui propose la structuration de la vidéo à trois niveaux différents. D'abord, le niveau des paragraphes vidéo (équivalent des micro-plans) est déterminé par l'analyse conjointe de la vidéo, des silences dans le flux

audio et de la transcription automatique de la parole. Ensuite, les frontières de plans sont déterminées par l'analyse de l'histogramme couleur et du flot optique. Enfin, des images-clé sont choisies parmi les images statiques au sens de l'analyse du flot optique.

VideoQ [32] est un outil de recherche dans une collection de plans vidéo. Dans chaque image, un ensemble de régions est défini comme un ensemble de pixels homogènes en termes de couleur, de texture ou de forme. Un objet vidéo est défini par un ensemble de régions groupées selon certains critères au cours de plusieurs images. VideoQ caractérise les objets par leurs couleurs, forme, texture et le mouvement. Le mouvement est central dans ce système qui propose une interface utilisateur permettant de définir une requête à l'aide d'un croquis des objets (forme, couleur et texture) intégrant leur trajectoire.

En plus de la segmentation automatique d'un flux vidéo en plans, micro-plans et objets, l'extraction automatique de texte peut être utilisée pour compléter l'analyse du flux vidéo. Par exemple, dans le projet Informedia [75], un système de transcription automatique de la parole (TTS) est utilisé pour produire des mots-clé associés aux paragraphes. Les mots-clés peuvent être utilisés pour accéder aux plans vidéo associés.

La reconnaissance automatique des textes incrustés dans le flux vidéo [162] apporte une information supplémentaire qui est utilisée pour raffiner l'indexation des segments vidéo (identification des lieux ou des locuteurs).

Le groupement d'un ensemble de plans partageant le même contenu sémantique (unité de lieu, de temps ou d'action), requiert un critère permettant de grouper les plans en scènes. Ce critère peut être basé sur l'analyse du flux audio pour grouper ensemble les plans partageant les mêmes caractéristiques audio [128, 138, 148]. Dans [176], c'est la similarité du contenu des plans, basée sur la couleur, conjointement à la proximité temporelle, qui permet de définir des "story units". Dans [150], le groupement de plans en scènes obéit à des règles issues de la production cinématographiques. Ces règles sont utilisées lors du montage afin de conserver une cohérence dans l'esprit du spectateur (respect de l'orientation et des lieux). Ces règles ("180° system", "shot/reverse shot" et "establishment/breakdown/re-establishment") permettent de définir des types de scènes.

Dans VideoManga [158], chaque plan d'un document vidéo est caractérisé par une image-clé et l'ensemble des images-clé est arrangé à la manière d'une bande dessinée. La taille de l'image indique l'importance du plan dans le document vidéo. L'importance de l'image-clé associée à un plan est proportionnelle à sa durée. La figure 1.4 présente la vue globale proposée à l'utilisateur. Elle donne un aperçu immédiat de la totalité du document vidéo ainsi que de ses plans importants.

FIG. 1.4: Résumé d'un document vidéo avec VideoManga.

1.1.3.2 Consultation d'un document vidéo

Le parcours linéaire du document vidéo utilise essentiellement la métaphore du magnétoscope, le document peut être parcouru image par image, en lecture et en avance rapide. Ce type d'interface possède également une "timeline", un intervalle représentant la durée du document muni d'un curseur graphique qui matérialise la position courante dans le

FIG. 1.5: Mosaïque spatio-temporelle.

document vidéo. Les principaux lecteurs multimedia du marché proposent ce type d'interface : helix player, quicktime player, xmms, winamp, windows media player, videolan, etc.

Le chapitrage automatique des plans et scènes d'un document vidéo permet un accès non-linéaire au segment recherché. Dans VideoStar [77], une hiérarchie de descriptions textuelles correspondant à la décomposition d'un document vidéo en scènes et plans est présentée à l'utilisateur comme point d'entrée dans le document vidéo.

Dans VideoManga [158], l'ensemble des images-clé est affichée à l'utilisateur et chaque image-clé sert de point d'entrée dans le document.

Quand le nombre d'images-clé à afficher est trop important, une hiérarchie d'images-clé peut être utilisée pour décrire le document vidéo avec différents niveaux de granularité.

Dans [94], une mosaïque spatio-temporelle, construite en superposant l'ensemble des "frames" issues d'un plan dans le même repère, permet de résumer par une image fixe le contenu d'un plan entier. La figure 1.5 présente un exemple de mosaïque spatio-temporelle construite à partir d'un plan panoramique vertical (réalisée par l'équipe TEMICS de l'IRISA).

1.1.3.3 Formulation des requêtes

La formulation des requêtes dans des bases de données multimédia requiert des interfaces spécifiques permettant d'exprimer les caractéristiques des images ou des segments vidéo recherchés.

Des interfaces textuelles peuvent être proposées quand une indexation par mot-clés est disponible comme c'est le cas dans le système Informedia [75] ou VideoStar [77].

Dans WebSEEk [145], les images et les documents vidéo présents sur le Web sont organisés de manière automatique dans une hiérarchie de catégories. Cette hiérarchie fournit un point d'entrée dans la collection qui permet ensuite d'étendre la recherche au moyen d'interfaces de recherche basées sur le contenu. Il est alors possible d'effectuer une recherche en utilisant les caractéristiques couleur des images en modifiant l'histogramme des couleurs d'une image de référence.

L'interface utilisateur de VideoQ [32] permet de spécifier les caractéristiques de couleur, de forme, de texture et de trajectoire d'un objet recherché dans une séquence vidéo.

1.1.3.4 Bouclage de pertinence

Dans WebSeek [145], l'utilisateur a la possibilité de raffiner la recherche en cours en affectant des poids positifs ou négatifs aux éléments de la liste de résultats courante. Une nouvelle liste est alors produite en tenant compte de la pondération fixée par l'utilisateur.

Dans [110], une interface hiérarchique permet à l'utilisateur de parcourir une collection d'images indexées grâce à différents descripteurs MPEG-7. La mesure de dissimilarité utilisée est paramétrée par un ensemble d'exemples positifs sélectionné par l'utilisateur à chaque étape de la recherche.

1.1.3.5 Produits commerciaux

QBIC [53] est un produit d'IBM dédié à la recherche dans de grandes collections d'images. En plus de recherches textuelles, QBIC permet à l'utilisateur d'effectuer des requêtes à l'aide de croquis, par la description des couleurs, formes et textures et à l'aide d'images données en exemple. QBIC permet également de traiter les données vidéo mais sans tenir compte de la dimension temporelle, les caractéristiques utilisées sont les mêmes que celles utilisées pour les images.

VideoLogger [162] est un produit de la société Virage qui permet de segmenter un flux vidéo en plans, de générer un "storyboard" d'images-clé et d'analyser le contenu vidéo à l'aide de "plugins" d'analyse de la parole, de détection de visages, de textes ou de types audio (parole, silence, bruit). La société Convera propose un produit similaire intitulé ScreeningRoom [37].

1.1.4 Standard MPEG7

En février 2002, le consortium "Moving Picture Experts Group" (MPEG) a publié un nouveau standard intitulé "Multimedia Content Description Interface (MPEG-7). Un résumé de ce standard est proposé dans [109].

Le standard MPEG-7 propose un ensemble de descripteurs (D) et de schémas de description (DS). Les descripteurs sont constitués de mesures quantitatives de caractéristiques extraites d'un flux audio-visuel. Les schémas de description définissent la structure des descripteurs et leurs relations. Les meta-données MPEG-7 permettent d'indexer et d'effectuer des requêtes sur les contenus audiovisuels. Les contenus audio-visuels peuvent inclure des images, des modèles 3D, des flux audio et vidéo ainsi que des informations sur leur composition.

Le standard MPEG-7 normalise la manière de décrire les données multimedia mais ne concerne pas les techniques d'extraction des caractéristiques audiovisuelles ou les méthodes d'indexation et de requête. L'objectif de ce standard est essentiellement de garantir l'interopérabilité des systèmes utilisant l'indexation multimédia.

1.1.4.1 Modèles de couleur

Dans cette section, nous présentons les modèles de couleur utilisés dans la suite de cette thèse.

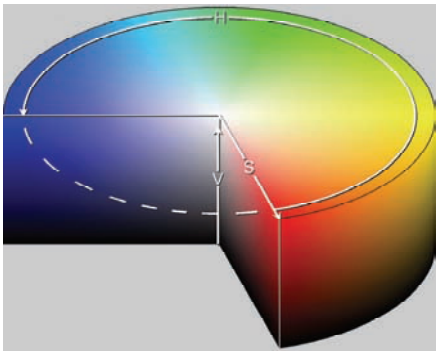
Le modèle de couleur utilisé pour le codage informatique des couleurs est le modèles RVB. Dans ce modèle, une couleur est obtenue par le mélange de chacune des trois composantes rouge, verte et bleue. L'espace de couleur RVB peut être représenté par un cube.

Le modèle de couleur YCbCr a été introduit dans le cadre des standards MPEG-1, -2 et -4 pour la représentation des couleurs dans le cadre du codage vidéo. La composante

Y désigne la luminance, la composante Cb désigne la composante bleue et la composante Cr désigne la composante rouge de la couleur codée. Les valeurs des trois composantes s'obtiennent à partir d'un triplet (R,V,B) en appliquant la formule de conversion suivante :

$$Y = 0.299.R + 0.587.V + 0.114.B \quad (1.4)$$
$$Cb = -0.169.R - 0.331.V + 0.500.B \quad (1.5)$$
$$Cr = 0.500.R - 0.419.V - 0.081.B \quad (1.6)$$

Le modèle TSL est également composé de trois composantes correspondant respectivement à la teinte, exprimée sous la forme d'un angle de 0 à 360°, d'une valeur de saturation comprise entre 0 et 1 indiquant la pureté de la couleur et d'une valeur de luminance, également comprise entre 0 et 1, qui indique si la couleur est claire ou sombre. L'espace TSL, (ou HSV pour "Hue, Saturation, Value" en anglais) se représente souvent sous la forme d'un cylindre (cf. figure 1.6).



a)

FIG. 1.6: Représentation du modèle de couleur TSL. ©Wikipedia.

1.1.4.2 Exemples de descripteurs de couleur

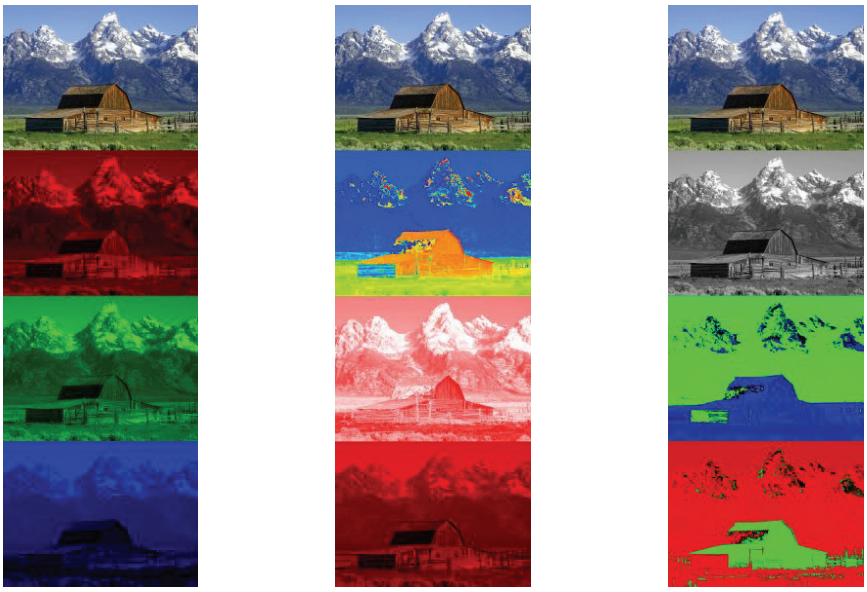
Nous présentons deux descripteurs de couleur MPEG-7. Nous y ferons référence à plusieurs reprises dans la suite de cette thèse.

Color Layout Descriptor (CLD) Ce descripteur représente la distribution des couleurs d'une image dans un format très compact. Cette compacité rend la mise en correspondance de ce descripteur avec d'autres descripteurs du même type très efficace.

La figure 1.8 illustre le processus d'extraction du CLD à partir d'une image en couleur. La résolution de l'image est réduite afin d'obtenir une image de 8×8 pixels. Chaque composante couleur de l'image (dans l'espace couleur $YCbCr$) subit ensuite une transformée DCT ("Discrete Cosine Transform", cf. Définition 3) qui produit 64 coefficients correspondant à la composition de différentes fréquences spatiales.

Définition 3 (DCT)
Soit une matrice de taille $N \times N$. Les coefficients de la transformée en cosinus discrète sont donnés par la formule suivante :

$$DCT(i,j) = \frac{1}{\sqrt{2}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x,y) cos\left(\frac{(2x+1)i\pi}{2N}\right) cos\left(\frac{(2y+1)j\pi}{2N}\right)$$



a) b) c)

FIG. 1.7: Illustration des modèles de couleurs RVB, TSL et YCbCr. a) Modèle RVB avec illustration de la composante rouge, verte et bleue (de haut en bas). b) Modèle TSL avec illustration de la composante de teinte, saturation et luminance (de haut en bas). c) Modèle YCbCr avec illustration de la composante de luminance, chrominance bleue et chrominance rouge (de haut en bas). ©Wikipedia.

avec $C(x) = \frac{1}{\sqrt{2}}$ *si* x *vaut* 0, *et* 1 *si* $x > 0$.

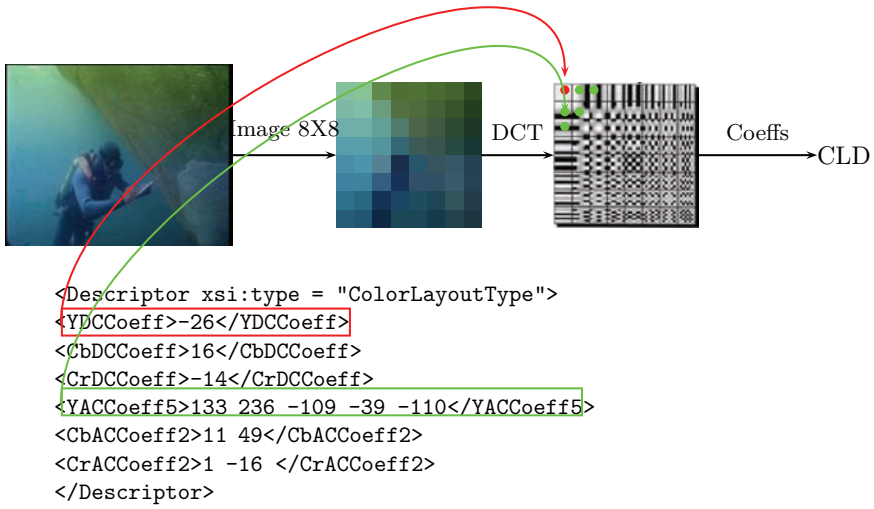


FIG. 1.8: Processus d'extraction du CLD.

Le coefficient $(0, 0)$, également appelé coefficient DC, (représenté par un point rouge sur la figure 1.8), correspond à la valeur moyenne du signal dans l'image. Les coefficients suivants selon l'ordre indiqué sur la figure 1.9 correspondent aux coefficients AC associés

aux fréquences moyennes. Ils correspondent aux régions significatives de l'image. L'ordre indiqué correspond donc à l'importance visuelle de la composante fréquentielle selon la perception humaine.

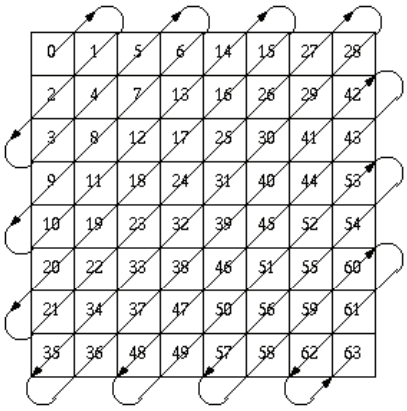
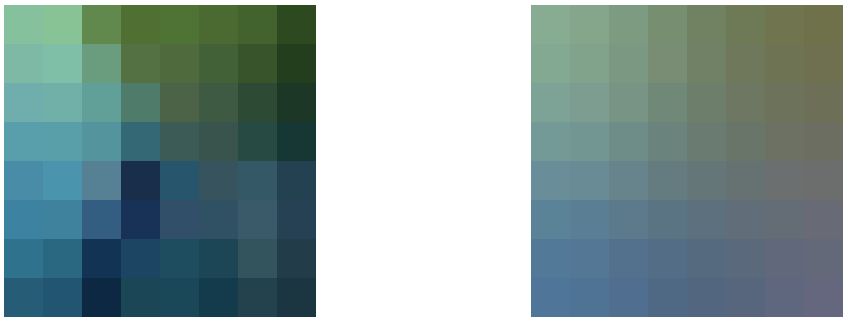


FIG. 1.9: Parcours "en zigzag" des coefficients fréquentiels des basses fréquences vers les hautes fréquences.

La figure 1.10 présente l'image reconstruite à partir de l'information stockée dans le CLD (par transformée inverse des coefficients). On remarque la disparition de certains détails par rapport à l'image originale liée à la suppression de coefficients fréquentiels.



a) b)

FIG. 1.10: Influence du nombre de coefficients fréquentiels dans le descripteur CLD. a) Version en résolution 8×8 de l'image originale. b) Image reconstruite à partir des coefficients du CLD (6 coefficients fréquentiels pour la luminance et 3 coefficients fréquentiels pour chaque composante couleur).

Une caractéristique intéressante du CLD est qu'il est possible de comparer des descripteurs issus d'images dont les tailles sont différentes. La précision de la description stockée peut être choisie en réglant le nombre de coefficients retenus dans le descripteur. Pour une utilisation avec des images fixes, le standard préconise de conserver 18 $(6 + 6 + 6)$ coefficients sur 192 $(3 * 64)$.

Les standard MPEG-7 indique également la mesure de dissimilarité associée au CLD.

Définition 4 (Mesure de dissimilarité du CLD)
Soit m, resp. n, le nombre de coefficients fréquentiels de luminance, resp. de chrominance. Soient $CLD_1 = ([Y_{1,i}]_{1 \leq i \leq m}, [Cb_{1,i}]_{1 \leq i \leq n}, [Cr_{1,i}]_{1 \leq i \leq n})$ et

$CLD_2 = ([Y_{2,i}]_{1\leq i\leq m}, [Cb_{2,i}]_{1\leq i\leq n}, [Cr_{2,i}]_{1\leq i\leq n})$ deux descripteurs CLD. La mesure de dissimilarité entre deux CLD, notée $\delta_{CLD} : CLD \times CLD \rightarrow \mathbb{R}^+$ est

$$\begin{aligned}\delta_{CLD}(CLD_1, CLD_2) &= \sqrt{\sum_i^m w_{y,i}(Y_{1,i} - Y_{2,i})^2} \\ &+ \sqrt{\sum_j^n w_{Cb,j}(Cb_{1,i} - Cb_{2,i})^2} \\ &+ \sqrt{\sum_j^n w_{Cr,j}(Cr_{1,i} - Cr_{2,i})^2}.\end{aligned} \tag{1.7}$$

Le système visuel humain étant plus sensible aux basses fréquences [15], la mesure de dissimilarité utilise une pondération plus importante pour le coefficient correspondant à la plus basse fréquence, appelé "coefficient DC" (les poids correspondants sont $w_{y,1}, w_{Cb,1}$ et $w_{Cr,1}$).

Dominant Color Descriptor (DCD) Ce descripteur fournit une description compacte des couleurs dominantes de l'image ainsi que leurs proportions. Le choix des couleurs dominantes est produit par un algorithme de fragmentation dans l'espace de couleur utilisé.

Définition 5 (Descripteur DCD)
Soit I une image, soit n le nombre de fragments couleur dans I, soit $(c_i)_{1\leq i\leq n}$ les centroïdes des fragments de couleur, $(p_i)_{1\leq i\leq n}$ les pourcentages des pixels de I associés au fragment d'indice i, $(v_i)_{1\leq i\leq n}$ les variances des fragments d'indice i et s la cohérence spatiale globale de l'image (définie par le nombre moyen de pixels $4-$connexes avec des pixels du même fragment de couleur). Le DCD de I est défini par

$$DCD = ((c_i, p_i, v_i)_{1\leq i\leq n}, s).$$

Définition 6 (Mesure de dissimilarité du DCD)
Soient $DCD_1 = ((c_{1,i}, p_{1,i}, v_{1,i})_{1\leq i\leq m}, s_1)$ et $DCD_2 = ((c_{2,j}, p_{2,j}, v_{2,j})_{1\leq j\leq n}, s_2)$ deux descripteurs DCD. Soit T_d la distance minimale séparant deux couleurs considérées comme similaires. Soit $a_{k,l}$ le coefficient de similarité entre deux couleurs défini par :

$$a_{k,l} = \begin{cases} 1 - \frac{||c_{1,k} - c_{2,l}||}{1.5.T_d}, & si\ ||c_{1,k} - c_{2,l}||\ \leq T_d \\ 0, & sinon \end{cases}$$

La mesure de dissimilarité entre deux DCD, notée $\delta_{DCD} : DCD \times DCD \rightarrow \mathbb{R}^+$ est

$$\delta_{DCD}(DCD_1, DCD_2) = \sum_{i=1}^m p_{1,i}^2 + \sum_{j=1}^n p_{2,j}^2 - \sum_{i=1}^m \sum_{j=1}^n 2a_{i,j}p_{1,i}p_{2,j}.$$

1.1.5 Évaluation des méthodes d'indexation

L'évaluation des performances des méthodes d'indexation est un point délicat. D'une part, l'accès à des collections d'images et de documents vidéo est limité par les droits intellectuels liés à ces œuvres. L'accès à des corpus de grande taille peut se faire via

des sociétés commerciales telles que Corel. Dans le cadre de cette thèse, la plupart des documents vidéo utilisés sont issus du corpus du centre de ressources et d'information sur les multimédias pour l'enseignement supérieur (CERIMES, ex SFRS). Le fait que les corpus ne soient pas disponibles librement rend la comparaison des méthodes d'indexation plus difficile.

D'autre part, l'évaluation repose souvent sur une "vérité terrain", c'est à dire une annotation parfaite d'un ensemble de documents de référence. Ces méta-données sont souvent produites manuellement et servent de référence pour comparer le résultat de l'indexation automatique. Par exemple, les frontières de plans d'un flux vidéo et les effets de transition peuvent être détectés manuellement et utilisés pour évaluer les détecteurs automatiques. La qualité et la publicité des annotations de références sont également indispensables pour permettre à différents auteurs de comparer leurs contributions.

Le but de la conférence TREC Video [123], sponsorisée par le "National Institute of Standards and Technologies" (NIST) est d'encourager les travaux concernant la recherche d'information en fournissant de grandes collections de test, des méthodes d'évaluation uniformes ainsi qu'un débat permettant aux participants de comparer leurs résultats.

Par exemple, les tâches proposées en 2006 se décomposent en 4 catégories :

- Détection des changements de plans : consiste à identifier les frontières de plan et leur type (coupure nette ou graduelle).
- Détection de caractéristiques de haut-niveau : consiste à détecter des concepts sémantiques ("indoor/outdoor", "People", "Speech",...).
- Recherche d'information : consiste à produire une liste des plans vidéo répondant le mieux au thème recherché.
- Exploitation de "rushes" : consiste à permettre la structuration et l'exploration des "rushes" vidéo, c'est à dire le contenu vidéo brut issu des séances de tournage et contenant de nombreuses répétitions de la même prise, des coupures nettes entre plans et un son naturel.

Dans le contexte de la recherche d'information l'efficacité d'une procédure de recherche d'information se mesure grâce au rappel et à la précision.

Définition 7 (Rappel et précision)
Soit Q une collection de documents, soit $P \subseteq Q$ l'ensemble de documents pertinents pour une requête données et soit $R \subseteq Q$ l'ensemble de documents retourné par une méthode de recherche d'information. Le rappel ("Recall") est défini par :

$$Recall = \frac{|P \cap R|}{P}.100\%.$$

La précision ("Precision") est définie par :

$$Precision = \frac{|P \cap R|}{R}.100\%.$$

Pour mesurer la performance des détecteurs de transitions graduelles dans les frontières de plans, les mesures de "frame recall" et "frame precision" ont été utilisées dans le cadre de la conférence TREC Video [123].

Définition 8 ("Frame recall" et "frame precision")
Soit $\mathcal{F}^{GT}$ l'ensemble des images d'un document vidéo appartenant à une transition et soit $\mathcal{F}^{COMP}$ l'ensemble des images détectées comme appartenant à une transition par une méthode. La mesure de "frame recall" est définie par

$$FrameRecall = \frac{|\mathcal{F}^{GT} \cap \mathcal{F}^{COMP}|}{|\mathcal{F}^{GT}|}.100\%$$

La mesure de "frame precision" est définie par

$$FramePrecision = \frac{|\mathcal{F}^{GT} \cap \mathcal{F}^{COMP}|}{|\mathcal{F}^{COMP}|}.100\%.$$

Ces mesures ont été étendues pour évaluer la qualité des groupement de plans en scènes [18] .

Définition 9 ("Shot recall" et "shot precision")
Soit $\mathcal{P} = \{P_i\}_{1 \leq i \leq N}$ *l'ensemble des plans vidéo d'une séquence. Soit* $\mathcal{H} = \{H_q\}_{1 \leq q \leq Q}$ *la partition de* $\mathcal{P}$ *en* Q *scènes telles que* $\bigcup_{1 \leq q \leq Q} H_q = \mathcal{P}$ *et* $\bigcap_{1 \leq q \leq Q} H_q = \emptyset$. *Soit* $\mathcal{H}^{GT} = \{H_t^{GT}\}_{1 \leq t \leq T}$ *la partition de référence de l'ensemble de plans* $\mathcal{P}$ *en* T *scènes.*

Pour une scène calculée H_p^{COMP} *et la scène de référence associée* H_t^{GT}, *c'est à dire celle pour laquelle l'intersection est maximale, les mesures de "shot recall" et "shot precision" sont définies par*

$$ShotRecall = \frac{|H_t^{GT} \cap H_q^{COMP}|}{|H_t^{GT}|}.100\%$$

$$ShotPrecision = \frac{|H_t^{GT} \cap H_q^{COMP}|}{|H_q^{COMP}|}.100\%$$

Le rappel donne la proportion des plans d'une scène détectée qui font partie de la scène de référence associée. Il caractérise ainsi la capacité de détection. La précision indique le pourcentage de plans d'une scène détectée qui a été correctement affecté à cette scène.

1.2 Notions sur les graphes

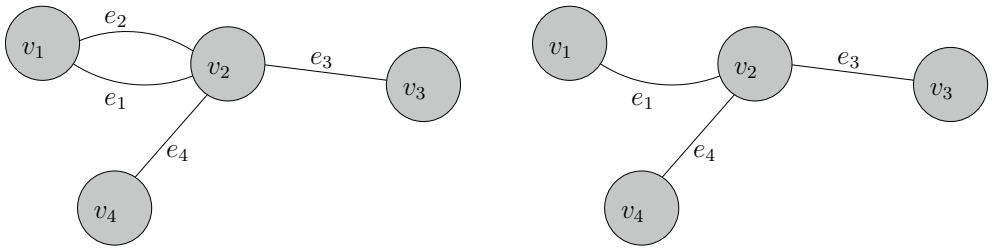
Dans cette section, nous introduisons les définitions et les notations sur les graphes que nous serons amenés à réutiliser dans la suite de cette thèse. Le lecteur peut également se référencer à différents ouvrages [20, 23].

Définition 10 (Graphe)
Soit V un ensemble d'objets appelés sommets. Soit $\mathcal{P}_2(V)$ l'ensemble des parties à deux éléments de V et soit $E \subseteq \mathcal{P}_2(V)$. On appelle graphe le couple (V, E). Les éléments de E sont appelés les arêtes du graphe.

On utilise habituellement la lettre V pour désigner l'ensemble des sommets d'un graphe à cause du mot anglais "vertex", de même pour l'ensemble des arêtes E qui vient du mot anglais "edge".

Un graphe comportant des *multi-arêtes* est appelé *multigraphe*. Des multi-arêtes sont constituées d'au moins 2 arêtes reliant la même paire de sommet. Une *boucle* est une arête reliant un sommet à lui-même. Un graphe est dit *simple* s'il ne contient pas de multi-arête ni de boucle.

Sauf mention contraire, dans cette thèse, le terme graphe fait référence à un graphe simple.



a) Un graphe avec 2 multi-arêtes (e_1 et e_2). b) Un graphe simple.

Fig. 1.11: Exemple de graphes.

Définition 11 (Arc)
Un arc *est une paire ordonnée de sommets :* $(u, v) \neq (v, u)$.

Définition 12 (Graphe orienté)
Soit $G = (V, E)$ un graphe. Un graphe est dit orienté si ses arêtes sont des arcs. Si $e = (u, v)$ est une arête, on transforme e en arête orientée en distinguant le sommet u, appelé source et noté $src(e)$, du sommet v appelé destination et noté $tgt(e)$.

Définition 13 (Adjacence d'un sommet)
Soient G un graphe orienté et $v \in V$. On appelle adjacence entrante, (resp. adjacence sortante, resp. adjacence), noté $adj^-(v)$, (resp. $adj^+(v)$, resp. $adj(v)$), l'ensemble des sommets $\{u|(u, v) \in E\}$, (resp. $\{u|(v, u) \in E\}$, resp. $\{u|(u, v) \in E\} \bigcup \{u|(v, u) \in E\}$).

Sur la figure 1.12, l'adjacence entrante de v_2 est l'ensemble de sommets $\{v_1, v_4\}$, son adjacence sortante est $\{v_3\}$.

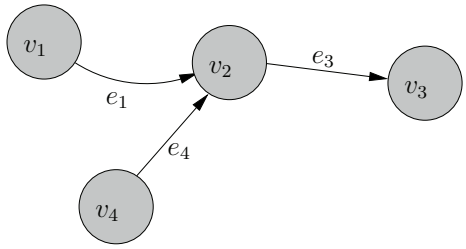


FIG. 1.12: Graphe orienté.

Définition 14 (Incidence d'un sommet)
Soient G un graphe orienté et $v \in V$. On appelle incidence entrante (resp. incidence sortante, resp. incidence), notée $inc^-(v)$, (resp. $inc^+(v)$, resp. $inc(v)$), l'ensemble des arêtes $\{(u,v)|(u,v) \in E\}$, (resp. $\{(v,u)|(v,u) \in E\}$, resp. $\{(u,v)|(u,v) \in E\} \bigcup \{(v,u)|(v,u) \in E\}$).

Sur la figure 1.12, l'incidence entrante de v_2 est l'ensemble d'arêtes $\{e_1, e_2\}$, son incidence sortante est $\{e_3\}$.

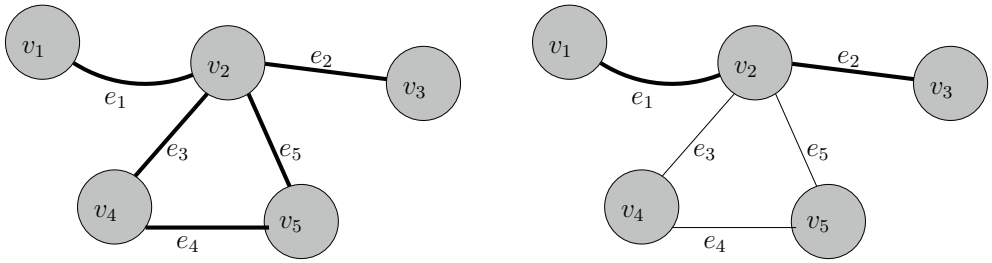
Définition 15 (Degré)
Soient G un graphe et $v \in V$. La cardinalité $|adj^-(v)|$ (resp. $|adj^+(v)|, |adj(v)|$) est appelée degré entrant (resp. degré sortant, degré) de v. On note cette cardinalité $deg^-(v)$ (resp. $deg^+(v)$, $deg(v)$).

On notera α le degré maximum de G

$$\alpha = \max_{v \in V}(deg(v)).$$

Le degré maximum du graphe de la figure 1.12 est égal à 3. C'est le degré du sommet v_2.

Définition 16 (Parcours)
Soit G un graphe, on appelle parcours de v_1 à v_k dans le graphe G, toute séquence $W = v_1, e_1, v_2, ..., v_{k-1}, e_{k-1}, v_k$ telle que $v_i \in V$, $e_i = (v_i, v_{i+1}) \in E$. La longueur du parcours est le nombre d'arêtes qui le composent. De plus, si pour tout v_i, v_i n'apparaît qu'une fois dans W, on appellera W un chemin.



a) Un parcours de v_1 à v_3 qui n'est pas un chemin. b) Un chemin de v_1 à v_3.

FIG. 1.13: Exemple de parcours de graphe.

Dans la figure 1.13, le parcours $W_1 = v_1, e_1, v_2, e_5, v_5, e_4, v_4, e_3, v_2, e_2, v_3$ n'est pas un chemin car le sommet v_2 y apparaît 2 fois. Le parcours $W_2 = v_1, e_1, v_2, e_2, v_3$ est un chemin.

La *longueur* d'un chemin $v_1, e_1, .., e_{k-1}, v_k$ est le nombre d'arêtes du chemin, c'est à dire $k-1$.

Définition 17 (Distance dans un graphe)
Soit $G = (V, E)$ un graphe et $u, v \in V$. La distance entre u et v dans G, notée $d_G(u, v)$, est la longueur du plus court chemin de u à v dans le graphe.

Définition 18 (Cycle)
Soit G un graphe. On appelle cycle tout parcours $W = v_1, ..., v_k$ tel que $v_1 = v_k$.

Dans la figure 1.13, le parcours $v_2, e_3, v_4, e_4, v_5, e_5, v_2$ est un cycle de longueur 3.

Définition 19 (DAG)
Soit G un graphe orienté. Si G ne contient aucun cycle, on dit qu'il est acyclique (ou encore que c'est un DAG pour "directed acyclic graph").

Le graphe de la figure 1.12 est un exemple de DAG.

Définition 20 (Arbre)
Un arbre est un graphe connexe à n sommets non orienté possédant $n-1$ arêtes.

Définition 21 (Profondeur d'un sommet)
Soit T un arbre, soit u un sommet de T. La profondeur de u dans T, notée $prof(u)$ est égale à la longueur de son chemin à la racine.

Définition 22 (Graphes connexe)
Soit G un graphe. On dit que G est connexe si et seulement si $\forall u, v \in V$, il existe un chemin de u à v.

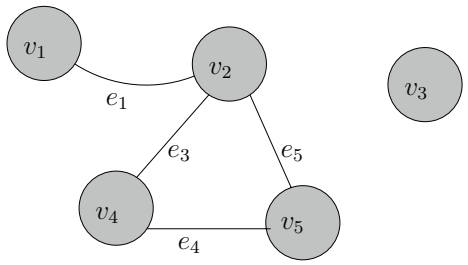


FIG. 1.14: Graphe non connexe.

Définition 23 (Sous-graphe induit)
Soit $G = (V, E)$ un graphe, soit $W \subset V$. Le sous-graphe induit de G par l'ensemble W est un graphe $G_W = (W, E_W)$ tel que $E_W = \{(x, y) \in E | x, y \in W\}$.

Définition 24 (Décomposition en composante connexe)
Soit G un graphe. On appelle décomposition en composante connexe, l'ensemble de sous-graphes $H = \{G_1, G_2, ..., G_k\}$ induit par la partition $\Pi = \{X_1, X_2, ..., X_k\}$ des sommets de G vérifiant les propriétés suivantes :
- *$\forall u, v \in X_i$, il existe un chemin de u à v,*
- *$\forall u \in X_i, v \in X_j, i \neq j$, il n'existe pas de chemin de u à v.*

Définition 25 (Graphe complet)
Un graphe $G = (V, E)$ complet est un graphe dont les sommets sont tous reliés deux à deux par une arête.

Définition 26 (Clique)
Soit $G = (V, E)$ un graphe. Une clique est un sous-graphe complet de G.

Définition 27 (Quasi-clique)
Soit $G = (V, E)$ un graphe, soit $W \subset V$, et G_W le sous-graphe induit par W. Un graphe est dit $\gamma-$dense, si $|E| \geq \gamma.\begin{pmatrix} |V| \\ 2 \end{pmatrix}$. Une $\gamma-$clique aussi appelée, quasi-clique de G, est un sous-graphe G_W de G induit par l'ensemble de sommets W qui est connexe et $\gamma-$dense.

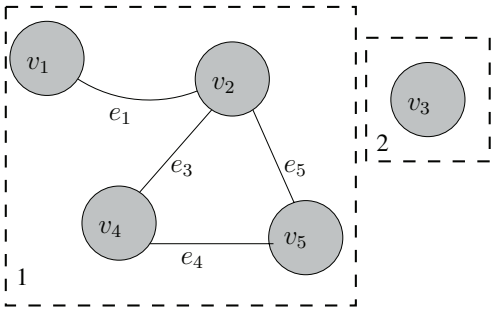


FIG. 1.15: La décomposition en 2 composantes connexes (1 et 2) du graphe de la figure 1.14.

Définition 28 (Graphe quotient)
Soient $G = (V, E)$ un graphe et $H = \{G_1, G_2, ..., G_k\}$ le multi-ensemble tel que $\forall i \in [1, k], G_i = (V_i, E_i)$ est un sous-graphe de G. On appelle graphe quotient de G par H le graphe $G_q(V_q, E_q)$. $V_q = \{v_1, v_2, ..., v_k\}$ désigne l'ensemble des sommets du graphe quotient.

Soit $\Phi : V_q \to H$ l'application qui associe à chaque sommet v_i le graphe $G_i = (V_i, E_i)$ (on note $\Phi_V(v_i)$ l'ensemble V_i). Soit $Q_{u,v}$ l'ensemble défini par :

$$Q_{v_i,v_j} = \{(u, v) | u \in \Phi_V(v_i), v \in \Phi_V(v_j), (u, v) \in E\}$$

On construit alors $E_q \subseteq V_q \times V_q$ de la façon suivante :

$$E_q = \{(v_i, v_j) | v_i \neq v_j, v_i \in V_q, v_j \in V_q, Q_{v_i,v_j} \neq \emptyset\}.$$

La figure 1.16 illustre la définition de graphe quotient. Les sous-graphes $G_1 = (\{v_1, v_2, v_4\}, \{e_1, e_2, e_3\})$ et $G_2 = (\{v_2, v_3, v_5\}, \{e_5, e_6, e_7\})$ sont les élément de l'ensemble $H = \{G_1, G_2\}$. Le graphe quotient $G_q = (V_q, E_q)$ est constitué de l'ensemble de sommets $V_q = \{v'_1, v'_2\}$ et par l'ensemble d'arêtes $E_q = \{(v'_1, v'_2))\}$.

Définition 29 (Valuation)
Soient G un graphe et $K \subset \mathbb{R}$. On appelle valuation des sommets (resp. des arêtes) du graphe, toute application $f_V : V \to K$ (resp. $f_E : E \to K$).

Définition 30 (Diamètre)
Soit G un graphe (valué ou non), le diamètre de G, noté $diam(G)$, est

$$diam(G) = max(\{d_G(u, v) | \forall u, v \in V\}).$$

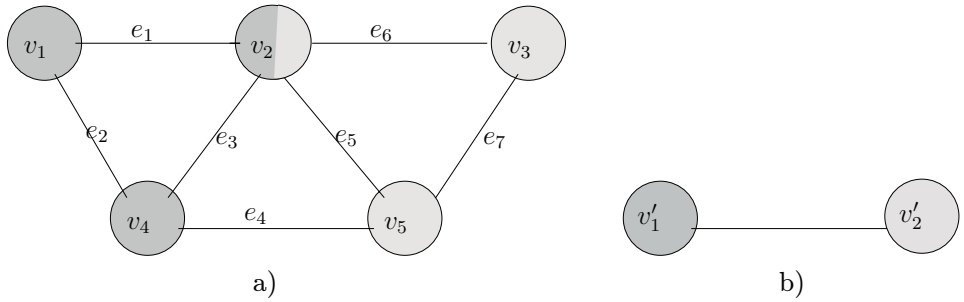


FIG. 1.16: Exemple de graphe quotient. a) Un graphe et 2 sous-graphes G_1 (sommets gris-foncés) et G_2 (sommets gris-clairs). b) Le graphe quotient associé.

Le diamètre du graphe de la figure 1.16 est égal à 2.

Définition 31 (Arbre orienté)
Soit G un graphe acyclique orienté connexe. G est un arbre si et seulement si $\forall v \in V$, $deg^-(v) \leq 1$.

Propriété :
Soit $T = (V, E)$ un arbre orienté et $v \in V$. Les propriétés suivantes sont vraies :
- $|E| = |V| - 1$.
- v est une feuille de l'arbre si et seulement si $deg^+(v) = 0$.
- v est l'unique racine de l'arbre si et seulement si $deg^-(v) = 0$. □

Le graphe de la figure 1.12 n'est pas un arbre car le sommet v_2 a un degré entrant supérieur à 1, $deg^-(v_2) > 1$.

L'arbre de la figure 1.17 possède une racine, le sommet v_1 et 3 feuilles, v_4, v_5 et v_6.

Définition 32 (Arbres k-aires)
Soit $T = (V, E)$ un arbre orienté. T est un arbre k-aire si $\forall v \in V, vois^+(v) \leq k$.

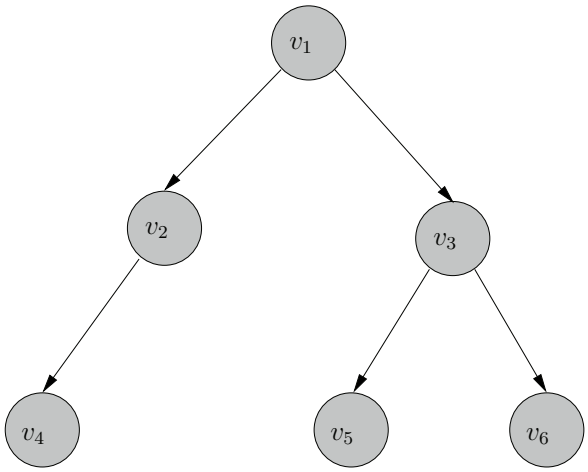


FIG. 1.17: Un arbre binaire (2-aire).

Définition 33 (Profondeur)
Soient $T = (V, E)$ un arbre et $s \in V$. On appelle profondeur de s, notée $prof_T(s)$, la longueur du chemin de la racine à s.

Dans l'arbre de la figure 1.17 le sommet v_2 a une profondeur de 1.

Définition 34 (Hauteur)
Soit $T = (V, E)$ un arbre. La hauteur h de T, notée $h(T)$, est :

$$h(T) = \max_{s \in V}(prof_T(s)).$$

L'arbre de la figure 1.17 a une hauteur égale à 2.

Définition 35 (c-spanner)
Soit $G = (V, E)$. Le c-spanner d'un graphe G est un sous-graphe G' de G dans lequel la distance entre toute paire de sommets $u, v \in V$ est au plus c fois leur distance dans G : $d_{G'}(u, v) \leq c.d_G(u, v)$ On appelle c le facteur d'étirement du spanner.

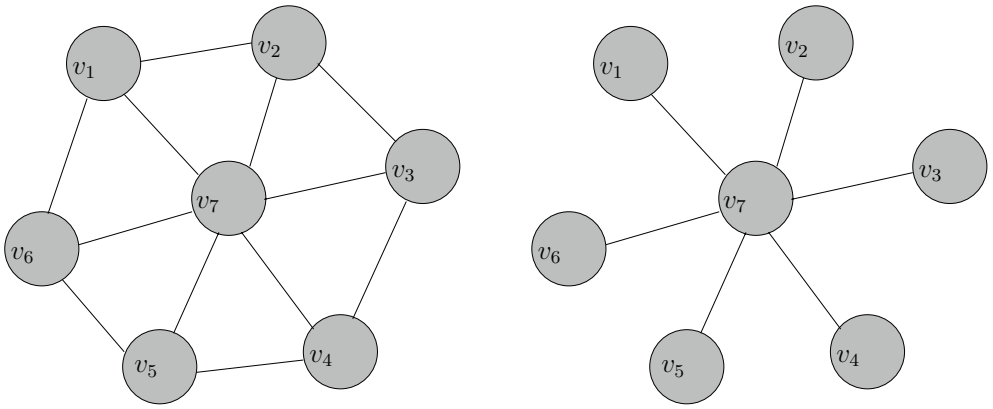


FIG. 1.18: Exemple de 2-spanner. a) Un graphe G. b) Le graphe G' qui est un 2-spanner de G.

Le graphe de la figure 1.18.b) est un 2-spanner du graphe de la figure 1.18.a) car pour toute paire de sommets, la distance qui les sépare dans G' est inférieure à deux fois leur distance dans G. Exemple : pour v_1 et v_6, on a $d_G(v_1, v_6) = 1$ et $d'_G(v_1, v_6) = 2$.

Définition 36 (Ensemble $k-$dominant)
Soit $G = (V, E)$, soit $S \subseteq V$ et soit $k \in \mathbb{R}^+$. L'ensemble de sommets S est $k-$dominant si $\forall v \in V$, $\exists u \in S$ tel que $d_G(u, v) \leq k$. On dit que u "couvre" v à distance k.

Définition 37 (Ensemble indépendant)
Soit $G = (V, E)$ et soit $S \subseteq V$. L'ensemble S est indépendant si $\forall v \neq v' \in S$, $s' \notin vois(s)$.

On appelle *$k-$séparation* la généralisation du concept d'ensemble indépendant aux graphes valués.

Définition 38 ($k-$séparation)
Soit $G = (V, E)$ un graphe valué, soit $S \subseteq V$ et soit $k \in \mathbb{R}^+$. L'ensemble S est$<$ une $k-$séparation si $\forall v \neq v' \in S$, $d_G(s, s') > k$.

Définition 39 (Problème des $k-$centres)
Soit $G = (V, E)$ un graphe valué et soit $S \subset V$ avec $|S| = k$. Le problème des $k-$centres consiste à trouver un ensemble $S \subset V$ de cardinal minimal tel que $\forall u \in V, d_G(u, S)$ soit minimal.

1.2.1 Plongement de graphes

Définition 40 (Plongement de graphe)
Soit G un graphe. On appelle fonction de plongement des sommets du graphe dans le plan Euclidien (plongement 2D) toute fonction $\rho : V \to \mathbb{R}^2$. On appelle plongement 2D des sommets de G, l'ensemble $P_G = \{(x_u, y_u) = \rho(u)\}_{\forall u \in V}$.

Définition 41 (Largeur d'un dessin)
Soit G un graphe et P_G un plongement de G en 2D. On appelle largeur de P_G, notée $l(P_G)$, la valeur

$$l(P_G) = max(\{x_u | (x_u, y_u) \in P_G\}) - min(\{x_u | (x_u, y_u) \in P_G\}).$$

Définition 42 (Hauteur d'un dessin)
Soit G un graphe et P_G un plongement de G en 2D. On appelle hauteur de P_G, notée $h(P_G)$, la valeur

$$h(P_G) = max(\{y_u | (x_u, y_u) \in P_G\}) - min(\{y_u | (x_u, y_u) \in P_G\}).$$

Définition 43 ("Aspect Ratio")
Soit G un graphe et P_G un plongement de G en 2D. L'"aspect ratio" du dessin, noté $A(P_G)$ est

$$A(P_G) = \frac{largeur(P_G)}{hauteur(P_G)}.$$

1.3 Visualisation d'information

La visualisation d'information est un domaine de l'informatique graphique qui traite de la représentation de *données abstraites* pour en faciliter la compréhension. C'est cette caractéristique qui distingue ce domaine de la visualisation scientifique.

La visualisation scientifique consiste à donner une représentation directe ou évidente des données. Par exemple, le rendu d'un modèle 3D d'organes issu de données d'IRM (Imagerie par Résonance Magnétique) ou encore l'affichage de coupes géologiques issues de données sismologiques.

Quand les données n'ont pas de structure géométrique naturelle et que de nouvelles conventions de représentation doivent être inventées, on s'éloigne de la visualisation scientifique pour entrer dans le domaine de la visualisation d'information.

Un exemple historique illustre l'importance de la représentation visuelle en tant qu'aide pour la compréhension et l'analyse de données abstraites. En 1954, une épidémie de choléra se déclara dans le quartier de Soho à Londres. Le docteur John Snow eut alors l'idée de représenter la position de chaque foyer atteint de choléra sur une carte du quartier. Il put ainsi établir un lien entre le point d'eau situé sur Broad Street et la concentration des cas de choléra (cf. figure 1.19).

FIG. 1.19: Carte originale du quartier de Soho, dessinée par le Dr. John Snow (1813-1858), où des points figurent les cas de choléra et les croix indiquent l'emplacement d'un point d'eau.

1.3.1 Métaphores visuelles

Les techniques de visualisation d'information produisent des vues sur les données traitées. Ces métaphores visuelles diffèrent selon la nature des données à représenter.

1.3.1.1 Données temporelles

Les données temporelles sont composées d'une dimension temporelle (l'instant ou la période) et d'une dimension structurelle (la donnée associée).

Dans le processus de visualisation, on distingue le cas où la dimension temporelle et structurelle sont fusionnées et le cas où elles sont traitées indépendamment [39]. La donnée structurelle est fusionnée à la donnée temporelle de manière explicite dans la représentation de données sous forme de courbe ou d'histogramme. La carte des pertes Françaises lors de la campagne de Russie (figure 1.20) créée par C.J. Minard et reprise par E. Tufte [156] est un exemple de représentation où la dimension temporelle est fusionnée mais représentée de manière implicite.

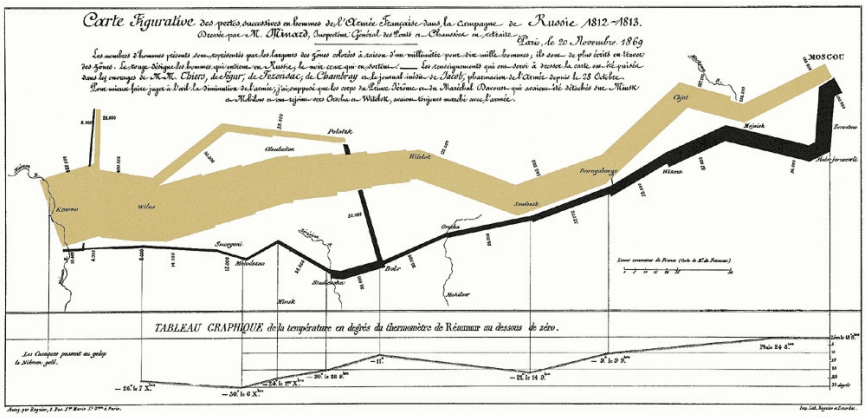


FIG. 1.20: Carte figurative des pertes de l'armée Française pendant la campagne de Russie 1812-1813.

Dans le cas où la dimension temporelle est traitée indépendamment de la dimension structurelle, on distingue sa représentation selon une ou plusieurs dimensions. Dans LifeLines [130], les auteurs proposent une interface pour la visualisation de dossiers médicaux ou de casiers judiciaires dans laquelle différents thèmes de la vie d'une personnes peuvent être affichés simultanément sous forme de frises chronologiques, facilitant ainsi la découverte de corrélations entre événements.

La représentation de la dimension temporelle en deux dimensions la plus courante est certainement le calendrier, qui fait ressortir la différence de granularité du temps (heures, jours, mois, ...). Les figures 1.22 et 1.23 sont issues des travaux de J. Van Wijk [160] sur l'identification de motifs et de tendances dans la consommation d'énergie et les heures de présence des employés dans une entreprise. Les différentes échelles de temps sont bien représentées grâce à l'utilisation de deux dimensions.

Certaines représentations mettent en avant la nature cyclique du temps. Dans [169], une représentation sous forme de spirale est proposée. La figure 1.24 représente l'intensité d'ensoleillement par jour.

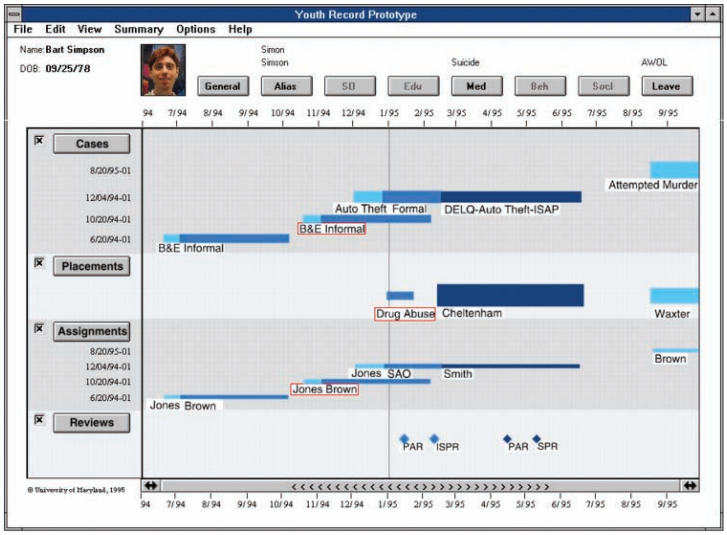


FIG. 1.21: Visualisation d'un casier judiciaire avec l'interface LifeLines.

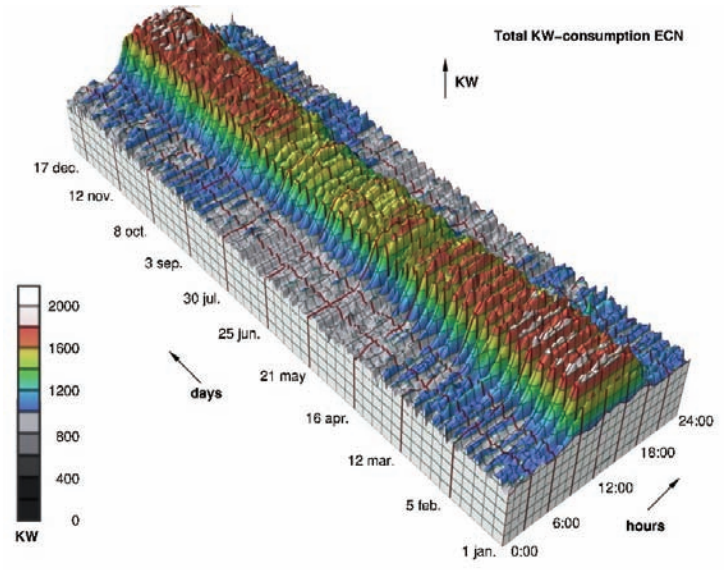


FIG. 1.22: Représentation de la consommation d'énergie en fonction de la date et de l'heure.

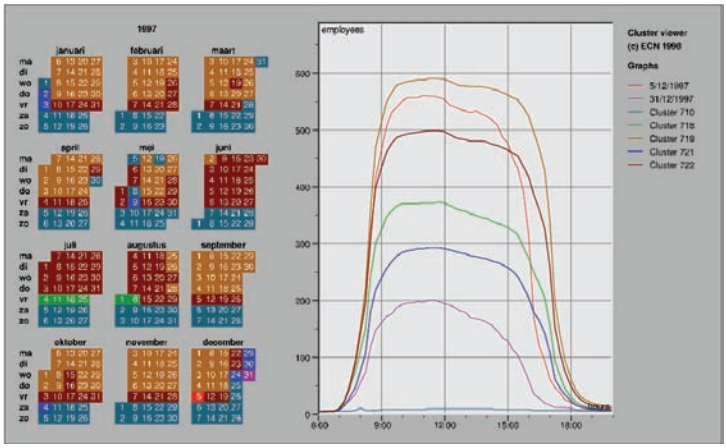


FIG. 1.23: Représentation des effectifs d'une entreprise selon 7 fragments caractéristiques des comportements des employés. On remarque le fragment associé à la veille de la Saint-Nicolas, le 5 Décembre et celui associé au 31 Décembre.

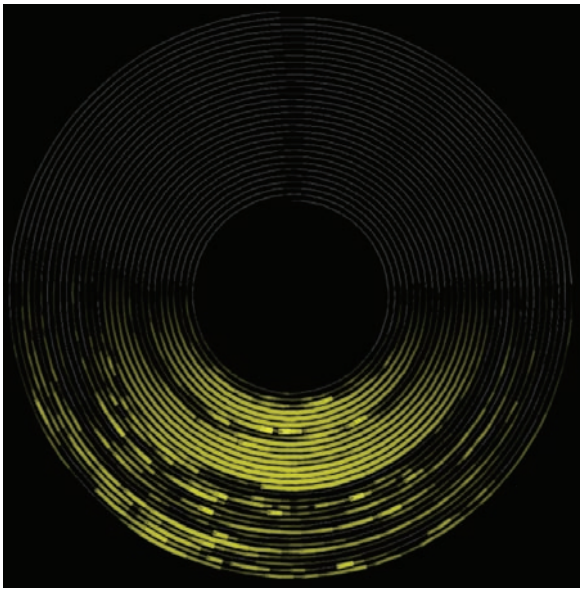


FIG. 1.24: Représentation de l'intensité d'ensoleillement sous forme de spirale.

La représentation de la dimension temporelle peut encore utiliser des techniques d'animation. Dans ce cas, la représentation de la dimension structurelle à un instant donné utilise la totalité d'une image. Le défilement des images représentant des instants différents produit une animation qui permet d'identifier les évolutions de la donnée structurelle.

1.3.1.2 Données multidimensionnelles

Données 1D Les données à une dimension correspondent à des listes d'objets comme les entrées d'un menu déroulant, un ensemble de documents, etc... Les métaphores visuelles proposées pour ce type de données sont nombreuses. On retrouve la métaphore du livre dans WebBook [28] afin d'organiser un ensemble de pages Web. La métaphore de la pile de documents est utilisée dans LifeStreams [56] pour l'organisation des données personnelles.

Données 2D Les données en deux dimensions correspondent aux diagrammes, données géographiques et topographies abstraites. Leur représentation naturelle revient à plaquer chaque couple de valeur sur une coordonnée du plan.

Données 3D Les données à trois dimensions correspondent à des données physiques, intrinsèquement 3D ou bien à la composition de données à deux dimensions auxquelles on associe un attribut supplémentaire.

Données nD Les données constituées de nombreuses dimensions posent des problèmes de visualisation dans la mesure où il n'existe pas de représentation naturelle de ces données. En fonction de la tâche à effectuer, le traitement des données multidimensionnelles sera différent. Si la tâche consiste à analyser les similarités, un pré-traitement permettant d'isoler les fragments peut être utilisé ou bien un algorithme de "multi-dimensionnal scaling" (cf. section 1.4) peut être appliqué pour produire un affichage en deux dimensions. Si la tâche consiste à analyser la corrélation existant entre les dimensions, alors une visualisation en coordonnées parallèles peut-être indiquée. En règle générale, l'exploration de données multidimensionnelles complexes fait appel à des techniques plus complexes qui intègrent l'interaction à différentes étapes du processus de visualisation (cf. figure 1.27).

1.3.1.3 Structures hiérarchiques

Les structures hiérarchiques sont naturellement présentes dans de nombreuses données : hiérarchies de personnes, systèmes de fichiers, arbres phylogéniques, fragments hiérarchiques.

De nombreux algorithmes de dessin automatique d'arbres ont été proposés par la communauté "graph-drawing" et tendent à maximiser des critères esthétiques tels que la longueur uniforme des arêtes, la répartition des sommets par niveau ou le dessin identique de sous-arbres isomorphes :

- Dessin hiérarchique de haut en bas proposé par Reingold et Tilford [134] et puis amélioré par Walker [165],
- dessin radial, produisant une meilleure occupation de l'espace disponible à l'écran et un placement central de la racine [51].
- dessin en ballon, produisant un dessin compact où les sous-arbres sont dessinés sous forme de disques [9].

Outre les menus arborescents présents dans la majorité des interfaces utilisateur, la visualisation d'informations hiérarchiques utilise des métaphores variées. Dans la représentation "Cone Trees" [136] les auteurs proposent la visualisation interactive d'une hiérarchie d'information en 3D suivant la métaphore du Rolodex^{TM} (cf. figure 1.25). Une autre utilisation de la 3D pour représenter des hiérarchies a été proposée par Rekimoto [135]. Cette représentation utilise des cubes emboités pour représenter un système de fichiers.

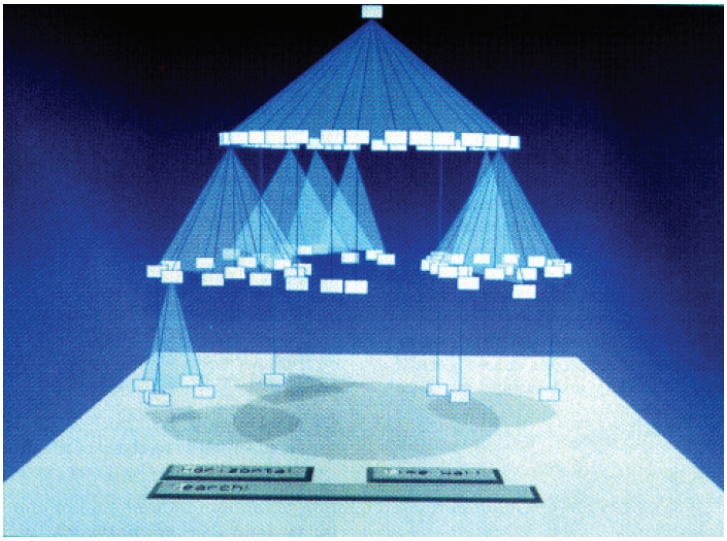


FIG. 1.25: Représentation d'une hérarchie d'informations avec un "ConeTree".

La visualisation basée sur les "TreeMaps" [84, 142, 14] utilise une subdivision de l'espace d'affichage en bandes horizontales et verticales pour représenter les ramifications d'une hiérarchie. Cette représentation est utilisée dans l'interface de visualisation des valeurs du marché boursier "marketmap" de SmartMoney [144]. Van Wijk et al. ont proposé un rendu graphique facilitant la lecture des "TreeMaps" [170].

1.3.1.4 Graphes

Les algorithmes de dessin de graphes se basent souvent sur certaines propriétés du graphe pour guider le dessin. Quand le graphe est orienté et sans-cycle (le graphe est un DAG), un ordre sur les sommets peut être choisi. Par exemple, le ou les sommets sans prédécesseurs peuvent être placés au sommet d'une représentation hiérarchique. Ce type de représentation doit minimiser le nombre de croisements d'arêtes entre les niveaux de la hiérarchie. L'algorithme proposé par Sugiyama et al. [147] est dédié aux dessin de DAG (Directed Acyclic Graph). Chaque sommet du DAG est associé à exactement une couche issue de la division de l'axe vertical du repère de dessin. Les arêtes reliant des sommets d'une même couche sont interdites et celles traversant plusieurs couches sont remplacées par une chaîne contenant un sommet virtuel par couche. Le calcul des abscisses des sommets est effectué en deux étapes. D'abord l'ordre des sommets dans chaque couche est modifié pour minimiser les croisements d'arêtes, ensuite le placement exact des sommets est calculé afin d'optimiser des critères esthétiques tels que le nombre de brisures sur chaque arête. Chacun de ces problèmes d'optimisation est NP-difficile et des heuristiques sont utilisées pour produire un résultat acceptable dans la plupart des cas.

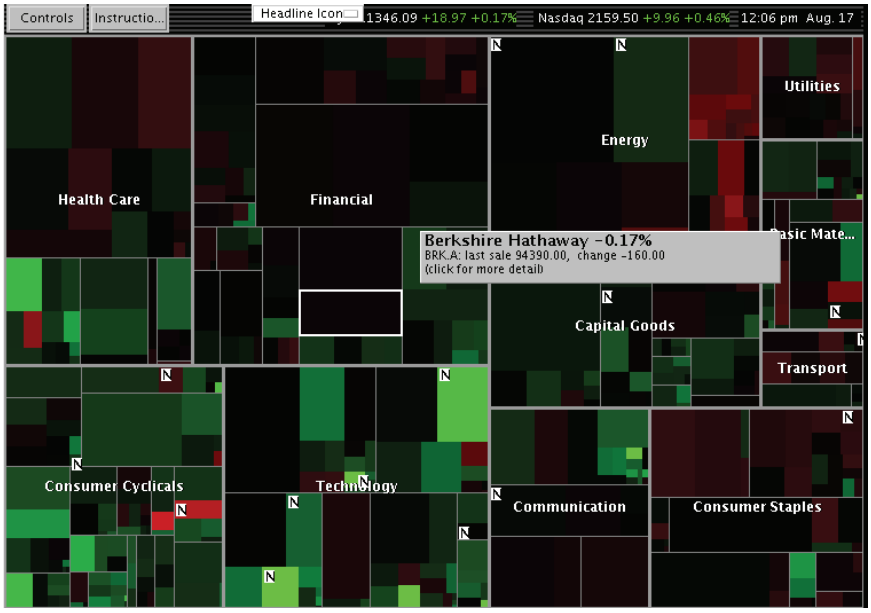


FIG. 1.26: "TreeMap".

Dans "Dig-Cola" [49] les auteurs utilisent un algorithme de dessin par modèle de force pour dessiner des graphes orientés. Ces méthodes, que nous présentons en détail dans la section 1.4, définissent une fonction de coût qui intègre des critères esthétiques associés au dessin du graphe. La minimisation de la fonction de coût produit un dessin de graphe satisfaisant selon les critères employés. Dans "Dig-Cola", la fonction de coût tient compte du degré de hiérarchie du graphe orienté. Cela permet de distinguer les partie hiérarchiques des parties non-hiérarchiques (cycles) dans le dessin produit. De plus, la méthode bénéficie des qualités des algorithmes de dessin par modèle de force telles que la représentation correcte des symétries et l'adéquation entre les relations de proximité existant dans le graphe et celles du dessin.

Une approche récente [3] concernant le dessin des graphes généraux consiste à rechercher des sous-graphes ayant une structure caractéristique, telle qu'une clique ou un arbre, et à représenter ces sous-graphes en utilisant le meilleur algorithme de dessin possible. Cette approche facilite la visualisation de ces structures caractéristiques dans le dessin final du graphe.

1.3.2 Techniques d'interaction

Les techniques de visualisation d'information incluent l'interaction de l'utilisateur avec les données. Cette interaction peut intervenir en différents points de la chaîne de visualisation des données dont les étapes sont présentées dans la figure 1.27.

1.3.2.1 Données

Le traitement préalable des données consiste à modifier le jeu de données à visualiser afin d'en faciliter la manipulation ou l'interprétation. Le jeu de données peut être filtré pour ne conserver qu'une partie des données initiales. Le jeu de données peut également

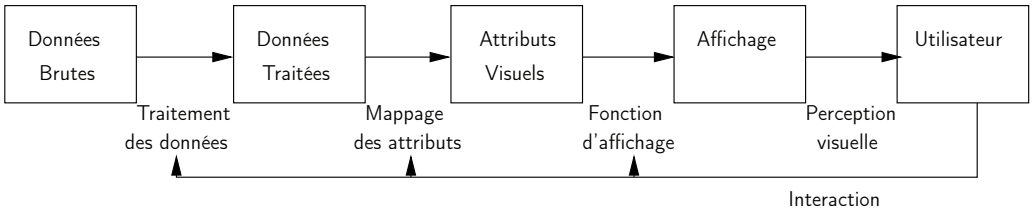


FIG. 1.27: Chaîne de visualisation d'information.

être structuré d'avantage. Par exemple, le calcul d'une relation de similarité permet de construire un graphe valué qui peut être facilement représenté en utilisant un algorithme de dessin adapté.

Un traitement fréquent consiste à identifier des fragments dans les données (étape de "clustering"). Un fragment est un ensemble d'entités considérées comme similaires selon certains critères. Dans le contexte de la visualisation, la fragmentation permet de résumer un fragment par un élément représentatif et de réduire la quantité d'information à afficher. L'analyse des fragments permet aussi de comprendre la structure d'un jeu de données et de choisir une technique de visualisation adaptée.

1.3.2.2 Attributs visuels

Le choix des attributs associés aux données constitue un point d'interaction important dans le processus de visualisation. La modification interactive des attributs visuels associés aux données constitue un outil d'exploration et de compréhension de celles-ci.

On distingue deux types de données fondamentaux qui sont les entités, et les relations. Chaque type de données peut avoir des attributs de trois types.

- Un attribut *quantitatif* correspond à une valeur numérique telle qu'une somme d'argent ou une superficie.
- Un attribut *ordonné* associe un rang à une entité comme dans le classement des élèves d'une classe.
- Un attribut *nominal* associe une catégorie à une entité. La marque d'un constructeur est un attribut nominal pour une voiture.

L'étape de mappage des attributs consiste à associer un attribut visuel aux attributs des entités et relations associés aux données. Ces attributs visuels sont donnés par ordre croissant d'efficacité pour la représentation d'attributs quantitatifs [164] : couleur et densité, volume, surface, angle et pente, longueur, position.

1.3.2.3 Niveau du rendu

Les techniques d'interaction agissant sur le rendu graphique permettent de modifier le point de vue sur la donnée selon différents objectifs. Le grossissement ou "zoom" géométrique permet de rapprocher le point de vue d'une région d'intérêt dans la représentation des données. Une évolution du "zoom" géométrique est le "zoom" sémantique qui consiste à afficher des informations supplémentaires quand le point de vue s'approche d'une région d'intérêt [13, 131].

Une autre évolution du "zoom" est la technique de "focus+context" qui consiste à afficher l'information la plus importante au centre de la vue avec le plus de détails et d'espace. L'espace autour du point focal est utilisée pour afficher le contexte, c'est à dire le reste de la structure de donnée. De nombreuses techniques de déformation de l'espace

autour du point focal permettent de représenter la totalité de la structure de données dans un espace réduit. C'est le cas dans "perspective wall"[103], "fisheye view"[59], "bifocal lens"[27] ou "hyperbolic tree"[97].

La technique de "brushing and linking" facilite l'interaction avec les données en grande dimension. Différentes vues sur les mêmes données sont affichées simultanément (histogrammes ou nuages de points [12]). Les éléments sélectionnés dans une des vues ("brushing") sont mis en évidence dans les autres vues ("linking"), ce qui permet de mettre en évidence les corrélations et les tendances dans les données traitées.

La visualisation basée-pixel ("pixel-oriented visualization") [90] est intéressante dans le contexte de la visualisation de grandes quantités de données en grande dimension. Cette technique consiste à associer une zone d'affichage de la taille d'un pixel à chaque entité à afficher. Les attributs quantitatifs pouvant être associés à chaque entité sont limités à la position et à la couleur. Un jeu de données composé de plusieurs dimensions peut être analysé en allouant une surface égale à chaque dimension, ce qui permet d'identifier les fragments et les corrélations entre dimensions. Cette technique a été utilisée pour la visualisation de l'historique du cours quotidien de quatre cours boursiers sur une période de huit ans [90], soit un total de 64800 valeurs affichées simultanément.

1.4 "Multidimensional Scaling"

Le "Multidimensional scaling" (MDS) est un ensemble de techniques issues des statistiques et utilisées en visualisation d'information pour explorer les similarités et les dissimilarités d'un jeu de données. La donnée consiste en une matrice de similarité entre paires d'éléments et le résultat consiste en un plongement de chaque élément dans un espace de 2 ou 3 dimensions, mieux adapté pour l'affichage.

Par exemple, le MDS peut être utilisé pour la visualisation de données multivariées, représentées sous forme de points $\overrightarrow{u_i} = (u_{i1}, .., u_{ip})^T$ dans un espace à p dimensions. On cherche une configuration de points $\overrightarrow{x_i} = (x_{i1}, .., x_{iq})^T$ dans un espace de dimension $q < p$, telle que la distance $\|\overrightarrow{x_i} - \overrightarrow{x_j}\|$ soit aussi représentative que possible de la distance $\|\overrightarrow{u_i} - \overrightarrow{u_j}\|$ dans l'espace de grande dimension.

Le MDS est utilisé dans de nombreux domaines pour l'analyse visuelle de données. En analyse financière, les facteurs impliqués dans la faillite des sociétés ont été étudiés par MDS [107]. La psychométrique [155], utilise le MDS pour mesurer le jugement de la similarité entre objets telle qu'elle est perçue par le sujet. Le marketing applique le MDS pour la création de cartes perceptuelles des marques commerciales.

Joseph B. Kruskal [95] est à l'origine du premier algorithme de MDS. Il a également formalisé l'adéquation entre les distances dans la configuration de petite dimension et les mesures de proximité initiales. Soit (S, δ) l'espace métrique composé de l'ensemble des données $S = \{x_1, x_2, ..., x_n\}$, muni de la mesure de dissimilarité notée δ. On note δ_{ij} la mesure de dissimilarité entre les éléments x_i et x_j et d_{ij}, la distance entre les éléments x_i et x_j dans la configuration de petite dimension, le *Stress* de la configuration [96] est

$$Stress_{\delta,d} = \sqrt{\frac{\sum_i \sum_j [\delta_{ij} - d_{ij}]^2}{\sum_i \sum_j d_{ij}^2}} \tag{1.8}$$

On constate que dans le cas d'un plongement qui conserve parfaitement les distances, la valeur du *Stress* serait égale à 0. En effet, la quantité $\delta_{ij} - d_{ij}$ de la Formule 1.1 est nulle pour tous i et j. La valeur maximum du *Stress* n'est pas 1 car dans le cas où $\delta_{ij} > 2.d_{ij}$ pour tous i et j, on obtient une valeur supérieure à 1. Pour illustrer ce cas de figure, on peut imaginer le plongement en 2D des coordonnées 3D des 4 coins d'un carré de coté 1. Imaginons que le plongement produise un carré de coté 3, alors on obtiendrait une valeur de *Stress* supérieure à un.

Les techniques de MDS présentent de nombreux avantages par rapport aux techniques de visualisation de données multivariées telles que les glyphes [33] ou les coordonnées parallèles [83]. Notamment, les techniques de MDS sont indépendantes du nombre de variables constituant les données et du type de valeur associée à chaque variable (catégorique nominale ou ordinale, numérique continue ou discrète) puisque la seule information requise est la mesure de dissimilarité entre toute paire d'éléments. La représentation des données ne perd pas en lisibilité avec l'accroissement du nombre d'éléments dans la collection puisque chaque nouvel élément peut potentiellement venir renforcer l'identification d'un groupe dans la vue 2D ou 3D en augmentant la densité de la zone correspondante dans le plongement. La figure 4.5 illustre le fait que l'identification des fragments est renforcée par l'augmentation du nombre d'éléments affichés.

Analyse statistique L'analyse en composantes principales [127] (ACP) est une technique statistique qui permet de sélectionner les axes qui expliquent le mieux la dispersion du nuage de points composé de n réalisations conjointes de p variables. Dans le contexte

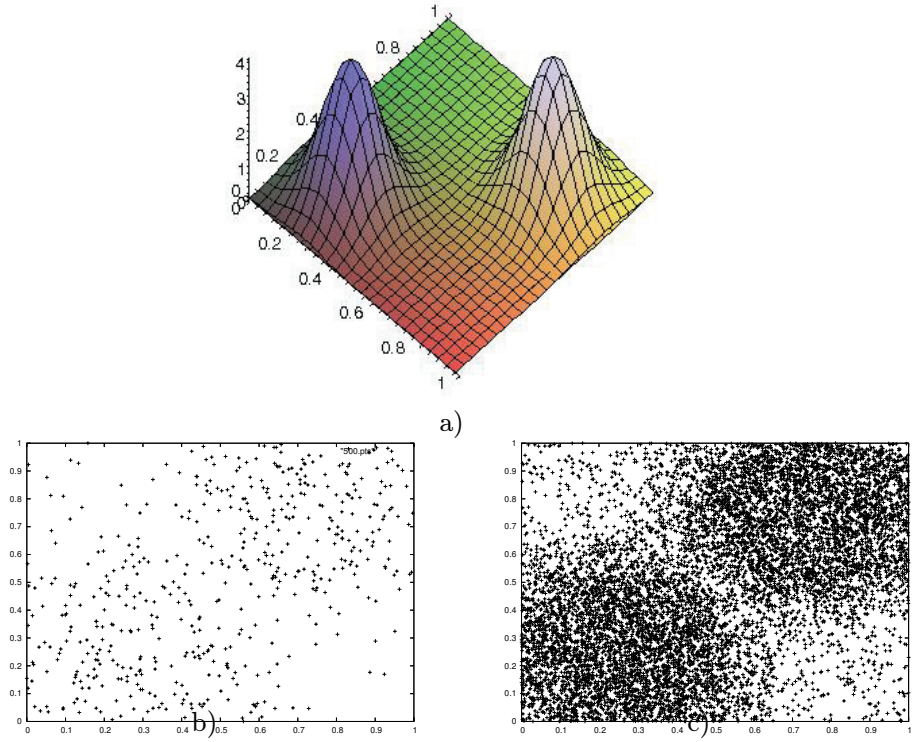


FIG. 1.28: L'identification des fragments est favorisé par la densité. a) Densité de probabilité de la position des points en 2D. 2 fragments sont modélisés par 2 lois normales $\mathcal{N}((0.25, 0.25), 0.2)$ et $\mathcal{N}((0.75, 0.75), 0.2)$. b) Visualisation de 500 points générés selon cette loi. c) Visualisation de 10000 points générés selon cette loi.

de la visualisation de données multivariées, on peut voir ce nuage de points comme un ensemble de n vecteurs numériques de dimension p. La projection des données dans le nouveau système de coordonnées composé des $q < p$ axes qui conservent le mieux la variance du jeux de données initial permet de visualiser ces données dans le cas où $q = 2$ ou $q = 3$. L'ACP détermine les q vecteurs propres de la matrice de covariance dont les valeurs propres sont les plus élevées. Le nouveau repère de projection est alors constitué des axes qui suivent les directions des vecteurs propres retenus.

Les limitations de l'ACP résident dans la nature des données qui peuvent être traitées (variables numériques continues ou discrètes) et dans le fait que la transformation appliquée aux données est linéaire. Intuitivement, cela revient à positionner un plan rigide 2D dans l'espace de grande dimension de telle sorte qu'il passe au plus près de l'ensemble des points. Pour un jeu de données tel que l'exemple de la figure 1.29, la projection des données sur un plan 2D par ACP ne permettrait pas de donner une représentation qui reflète la nature intrinsèquement 2D de la surface représentée par le nuage de points.

L'algorithme Isomap [151] apporte une solution à ce problème en modifiant la mesure de dissimilarité utilisée. Une approximation de la géodésique sur la surface formée par le nuage de points se substitue à la distance Euclidienne dans l'espace de départ. Cette

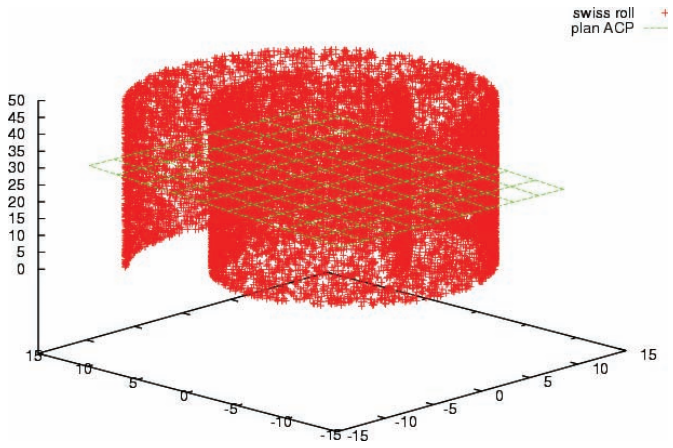


FIG. 1.29: Nuage de points en 3D, échantillonné à partir d'une surface enroulée en spirale.

approximation est donnée par la distance dans le graphe $G = (V, E)$ avec $V = X$ et $E = \{(x_i, x_j) | \delta_{ij} \leq \epsilon\}$, ϵ étant le seul paramètre à fournir à l'algorithme.

Dessins par analogies avec la physique C'est en cherchant à résoudre le problème du dessin de graphe dans sa formulation la plus générale qu'est née une famille d'algorithmes très populaires pour la visualisation de données. Les algorithmes de dessin de graphe basés sur des analogies avec la physique [50] permettent de dessiner un graphe sans considérer des propriétés pouvant guider la technique de dessin. Bien que la planarité, la régularité ou la structure arborescente du graphe ne soient pas prises en compte par ce type d'algorithme, les dessins produits parviennent à informer sur les caractéristiques intrinsèques du graphe. L'objectif recherché est que la distance entre toute paire de sommets dans le dessin soit proportionnelle à la distance du plus court chemin entre ces sommets dans le graphe. On peut chercher également à maximiser des critères esthétiques généraux (cf. section 1.3.1.4).

Le principe de ces algorithmes est de considérer les sommets du graphe comme des objets interagissant selon les lois de la physique. Par exemple, les sommets représentent des aimants soumis à une force de répulsion et les arêtes des ressorts tendant à rapprocher les sommets connectés. L'algorithme calcule une configuration stable du modèle, avec l'espoir que le dessin de graphe associé soit satisfaisant. Ces algorithmes se caractérisent par :

- le modèle d'énergie associé à une configuration de sommets,
- l'algorithme de calcul qui minimise l'énergie du système et fournit une configuration stable des sommets.

Kamada et Kawai [87] proposent un modèle de forces basé sur l'énergie potentielle du ressort. Soit k le coefficient de raideur d'un ressort de longueur naturelle l et de longueur réelle d, l'énergie potentielle, noté $\mathcal{E}_p$ est

$$\mathcal{E}_p = k(d - l)^2 \tag{1.9}$$

Les auteurs proposent de rendre la longueur idéale du ressort proportionnelle à la distance dans le graphe entre les sommets u et v. L'énergie globale du système est donnée par la somme des énergies potentielles existant entre toute paire de sommets pour une configuration donnée. Soit $G = (V, E)$ un graphe, soit $P_G = \{p_u = (x_u, y_u)\}_{u \in V}$ la configuration en 2 dimensions des sommets du graphe, soit $d_G(u, v)$ la distance dans le graphe

entre $u, v \in V$, soit l la longueur idéale d'une arête et c une constante de mise à l'échelle. L'énergie potentielle induite par l'action des ressorts associés aux arêtes, notée $\mathcal{E}$, est

$$\mathcal{E} = \sum_{(u,v)\in E} \frac{c}{d_G(u,v)}.(\|p_u - p_v\| - l.d_G(u,v)) \tag{1.10}$$

La méthode utilisée pour trouver une configuration correspondant à un minimum (local) de cette énergie est la méthode de Newton-Raphson.

Davidson et Harel [40], proposent un modèle d'énergie plus éloigné de la réalité physique qui intègre des forces associées à des critères esthétiques associés au dessin de graphe :

- Distribution des sommets : afin d'assurer une distribution uniforme des sommets sur l'aire de dessin, une composante inversement proportionnelle à la distance entre sommets est prise en compte dans le calcul de l'énergie globale. Ainsi, les sommets placés trop proches se repoussent ;
- Confinement : une composante inversement proportionnelle à la distance à la bordure du dessin, fait croître l'énergie globale quand des sommets s'approchent des limites du dessin ;
- Longueur d'arête : une composante pénalise les arêtes trop longues ;
- Croisement d'arêtes : les croisements d'arêtes contribuent à l'augmentation de l'énergie globale ;
- Distance sommet-arête : la distance minimale entre un sommet et chaque arête influe de manière inversement proportionnelle sur l'énergie. Ceci permet d'éviter le chevauchement d'une arête avec un sommet dont elle ne serait pas issue.

Une configuration qui minimise l'énergie globale est trouvée par une technique de recuit simulé. A partir d'une configuration initiale aléatoire, des configurations voisines obtenues en déplaçant un seul sommet de manière aléatoires sont testées, si une nouvelle configuration contribue à faire baisser l'énergie du système alors elle est conservée. Une variable modélisant la température du système contrôle l'action du hasard dans la création des nouvelles configurations. Lorsque la température diminue, la configuration évolue de plus en plus lentement.

Une autre approche dans le contexte du dessin de graphes par analogie avec un modèle physique consiste à simuler l'action des forces sur les objets "physiques" que sont les sommets du graphe. Dans les 2 exemples précédents un modèle d'énergie inspiré de la physique est défini mais l'analogie s'arrête à ce stade puisque les techniques employées ensuite pour atteindre une configuration minimale en terme d'énergie font appel à l'optimisation ou à des heuristiques qui s'éloignent du modèle d'interactions entre particules.

Dans [50], les auteurs proposent l'algorithme du Spring-Embbeder qui applique à chaque sommet $u \in V$ le vecteur de déplacement correspondant à la résultante des forces d'attraction et de répulsion issues de l'interaction de u avec les autres sommets. Cette méthode "simule" littéralement les forces utilisées dans le modèle. Les forces sont appliquées en limitant toutefois le calcul des forces d'attraction aux sommets adjacents du sommet courant et les forces de répulsion aux sommets non-adjacents. Cependant, la complexité en temps de cet algorithme est en $O(n^3)$ puisque le calcul du bilan des forces de chaque sommet implique une interaction avec tous les autres sommets, ce qui requiert un temps proportionnel à $O(n^2)$, et ces interactions sont répétées un nombre d'itérations généralement linéaire.

Cet algorithme a l'avantage d'être simple à mettre en œuvre et permet de visualiser l'animation du système soumis aux forces en cours de calcul. L'algorithme 1.1 illustre la simplicité de la méthode. La fonction $s(t)$ permet de limiter l'amplitude du déplacement effectué à chaque itération.

Données: $G = (V, E)$, Placement initial $P_G = \{p_u = (x_u, y_u)\}_{u \in V}$
Retourne: Plongement de V avec un faible Stress
$t \leftarrow 0$
$\mathcal{E} \leftarrow +\infty$
TantQue Non $Minimum\_Atteint(\mathcal{E})$ **Faire**
  Pour $v \in V$ **Faire**
  $F_v(t) \leftarrow \sum_{u\ :\{u,v\} \notin E} f_{rep}(p_u, p_v) + \sum_{u:\{u,v\} \in E} f_{spring}(p_u, p_v)$
  FinPour
  Pour $v \in V$ **Faire**
  $p_v \leftarrow p_v + s(t).F_v(t)$
  FinPour
  $t \leftarrow t + 1$
  $\mathcal{E} \leftarrow Calculer\_Energie()$
FinTantQue

Algorithme 1.1: Spring-Embedder

L'algorithme de Graph-Embedder [57], est basé sur le même principe que le Spring-Embedder. Il inclut notamment l'ajout d'une force gravitationnelle qui empêche les composantes déconnectées d'un graphe de se disperser arbitrairement et un mécanisme qui pénalise les mouvements inefficaces lors de la simulation. Les sommets soumis à des rotations et des oscillations sont ralentis.

Échantillonage local La complexité en temps des algorithmes basés sur le Spring-Embedder est en $O(n^3)$ dans sa formulation initiale. Une complexité de cet ordre interdit le passage à l'échelle et donc l'affichage efficace de graphes de très grande taille. Les contributions suivantes présentent les stratégies employées pour réduire le temps d'exécution des algorithmes basés sur la simulation d'un modèle de forces tout en conservant une qualité de dessin comparable.

Dans [58], les auteurs adaptent l'algorithme du Spring-Embedder, notamment pour en améliorer la complexité en temps, en limitant le calcul des forces de répulsion avec les sommets voisins, faisant passer la complexité en temps en $O(n^2)$. Pour accélérer la convergence du système, le calcul des forces est quadratique en fonction de la distance entre les sommets.

Échantillonage aléatoire Dans [31], l'auteur propose une stratégie qui permet de faire passer la complexité en temps d'une étape de calcul des forces entre sommets de $O(n^2)$ à $O(n)$, faisant ainsi passer le temps d'exécution global de l'algorithme en $O(n^2)$. A chaque itération, un sommet u n'interagit qu'avec les sommets des ensembles V_u et S_u qui sont de taille constante. L'ensemble V_u conserve les sommets les plus proches de u parmi les sommets tirés de manière aléatoire à chaque itération. L'ensemble S_u est l'échantillon aléatoire renouvelé à chaque itération. Les interactions de u avec S_u contribuent au placement global de u alors que les interactions de u avec V_u raffinent le positionnement de u à travers des interactions locales.

L'amélioration apportée dans [119] permet d'abaisser le temps d'exécution à $O(n\sqrt{n})$ utilisant un échantillon aléatoire, S, de taille $\sqrt{n}$. Cet échantillon est d'abord placé en utilisant l'algorithme précédent en $O(n^2)$, puis chaque sommet u, absent de l'échantillon

est placé en interagissant avec le sommet qui lui est le plus proche dans S, son "parent", et un sous-ensemble de taille constants de sommets de S. Le placement des sommets de S fournit une approximation du placement final en temps $O(n)$. Les autres sommets utilisent les sommets de S comme références pour leur propre placement. La recherche du parent dans S, avec $|S| = \sqrt{n}$, contribue à la complexité globale de l'algorithme en $O(n\sqrt{n})$.

Dans [118, 86], les auteurs améliorent la complexité en temps globale de l'algorithme de Morrison et al. [119] en réduisant le temps de recherche du parent.

Théorème 1.1 ([118])
La complexité en temps de l'algorithme de Morrison et al. [119] peut être réduite à $O(n^{1/4})$.

Échantillonage hiérarchique La qualité des plongements obtenus par les approches basées sur l'échantillonnage aléatoire présentées dans [119, 118, 86] sont très dépendantes de l'échantillon aléatoire S utilisé comme référence. Certains (mauvais) choix peuvent conduire à une sous-représentation de certaines parties du graphe et un grand nombre de sommets peut choisir un parent très éloigné dans le graphe. Les travaux qui suivent utilisent le mécanisme suivant :

- Construction d'une hiérarchie d'échantillons des sommets du graphe de plus en plus dense.
- Parcours de l'échantillon le moins dense vers le plus dense avec positionnement initial des nouveaux sommets et raffinement de la position des sommets déjà présents..

L'intuition derrière ces approches est d'utiliser une décomposition multi-échelle du graphe pour représenter de manière fidèle les relations globales et locales entre les sommets.

Dans GRIP [60], les auteurs utilisent un échantillonage hiérarchique des sommets du graphe qui garantit une bonne distribution des échantillons à chaque niveau. Soit $G = (V, E)$ un graphe, soit $\emptyset \subset V_k \subset ... \subset V_1 \subset V_0 = V$ une suite d'ensembles de sommets qui vérifient :

$$\forall u, v \in V_i,\ d_G(u, v) > 2^{i-1}. \tag{1.11}$$

Une fois l'échantillon hiérarchique produit, l'algorithme consiste à positionner les échantillons en partant du moins dense, vers le plus dense. Les sommets absents des échantillons déjà placés, se positionnent par rapport à un nombre constant de parents par niveau déjà traité en utilisant un calcul de forces similaire à celui de Kamada-Kawai [87]. C'est la recherche de ces parents qui domine la complexité globale de l'algorithme qui est en $O(|V|(log|V|)^2)$.

Dans FMS [73], les échantillons sont composés d'une hiérarchie de graphes $G^{k_1}, ..., G^{k_l}$, $k_1 < k_2 < ... < k_l = |V|$ qui sont des 2-approximation des k-centres du graphe initial. La phase de raffinement utilise la méthode de Kamada-Kawai [87]. L'algorithme nécessite de stocker la matrice de distance du plus court chemin entre toute paire de sommet (APSP), ce qui requiert un espace en $\Theta(|V|^2)$, ceci empêche le passage à l'échelle (cf. [70], Tableau 1 pour une vérification expérimentale).

Dans [69], les auteurs proposent, FM^3 (pour Fast Multipole Multilevel Method), un algorithme qui produit un plongement d'un graphe non-orienté pondéré en temps $O(|V| \log |V| + |E|)$. L'algorithme utilise la métaphore de "système solaire" pour décomposer le graphe de manière hiérarchique par contraction d'arête. L'originalité de la méthode concerne l'approximation des forces de répulsion qui agissent sur les sommets. Une décomposition de l'espace 2D, basée sur une structure de Quadtree optimisée, permet de restreindre le calcul des forces de répulsion des sommets d'une zone aux zones voisines. L'approximation des forces de répulsion est basée sur des outils de l'analyse complexe issues

de l'électromagnétique pour le calcul de l'énergie potentielle dans un système de particules chargées. Cette approche permet notamment de contrôler la précision de l'approximation de l'énergie et des forces qui agissent sur les sommets. Les auteurs proposent une comparaison expérimentale de leur méthode avec certains des algorithmes cités précédemment dans [70].

Synthèse La Tableau 1.1 synthétise les caractéristiques principales de chaque algorithme présenté dans ce chapitre.

Principe	Algorithme - Auteur	Complexité	Caractéristiques
Analyse statistique	ACP [127]	$O(n^3)$	
	Isomap, Tenenbaum et al. [151]	$O(n^3)$	– Approximation de la distance géodésique à la surface du nuage de points, – Utilisation de l'ACP pour le plongement en 2D.
Optimisation	Kamada-Kawai [87]	$O(n^2)$ par itération	– Basée sur une méthode de descente de gradient (Newton-Raphson).
	Davidson-Harel [40]	$O(mn^2)$	– Basé sur le recuit simulé. – Modèle d"énergie basé sur des critères esthétique (proximité arête-sommet, attraction du dessin au centre, espacement des sommets). – Dessin de graphe non pondéré.
Analogie physique	Spring-Embedder, Eades [50]	$O(n^3)$	– Forces linéaires.
	Fruchterman et Reingold [58]	$O(n^2)$	– Forces quadratiques pour accélérer la convergence, – Seul le voisinage proche donné par les cellules d'une grille sont utilisés pour f_{rep}, – $s(t)$ décroît avec t.
			suite page suivante

Principe	Algorithme - Auteur	Complexité	Caractéristiques
	GEM, Frick et al. [57]	$O(n^3)$	– Optimisation numérique dans le calcul des forces, – Introduction d'une force gravitationnelle, – $s(t)$ pénalise déplacements *inefficaces* (rotations et oscillations).
Echantillonage aléatoire	Chalmers [31]	$O(n^2)$	– A chaque itération, un nouvel échantillon aléatoire de taille constante est utilisé.
	Morrison et al. [119, 118]	$O(n\sqrt{n})$ [119] $O(n^{1/4})$ [118]	– Un échantillon de taille $\sqrt{n}$ est positionné en $O(n^2)$. – Placement des éléments restant en fonction du parent le plus proche s'effectue en $O(\sqrt{n})$ [119], resp. en $O(n^{1/4})$ [118].
	Jourdan et al. [86]	$O(n \log\ n)$	– Basé sur Morrison et al., la recherche du parent s'effectue en $O(\log n)$.
Echantillonage hiérarchique	GRIP, Gajer et al.[60]	$\Theta(n(\log n)^2)$	– Échantillonage hiérarchique du graphe initial (ensemble indépendant maximal), – Placement initial des sommets "intelligent".
	FMS, Harel et al.[73]	$\Theta(nm)$	– Échantillonage hiérarchique par approximation des k-centres, – Complexité en espace $\Theta(n^2)$.
	FM^3, Hachul et al.[69]	$O(n \log n + m)$	– Optimisation du calcul des forces de répulsion utilisant un Quadtree, – Approximation efficace des forces de répulsion.

TAB. 1.1: Tableau récapitulatif des principaux algorithmes de MDS.

L'algorithme de plongement d'un espace métrique dans le plan que nous proposons dans le chapitre 4 se base sur un échantillonage hiérarchique des données similaire à celui effectué dans GRIP [60].

1.5 Espaces métriques

Dans cette section, nous introduisons les notations et les définitions utilisées pour la manipulation des espaces métriques.

1.5.1 Notations

Nous introduisons les notations et les définitions de certains termes employés dans les chapitres suivantes.

Notation (Espace métrique) :
Soit (S, δ) un espace métrique définit sur l'ensemble $S = \{x_1, x_2, ..., x_n\}$ muni d'une fonction de distance entre éléments $\delta : S \times S \rightarrow \mathbb{R}^+$. □

Définition 44 (Boule ouverte)
Soient $u \in S$ et $r \in \mathbb{R}^+$, on appelle boule ouverte centrée en u et de rayon r, l'ensemble $B_u(r) = \{x \in S | \delta(x, u) < r\} \subset S$.

Définition 45 ("Aspect Ratio" d'un espace métrique)
Soit (S, δ) un espace métrique. L'"aspect ratio" de l'espace métrique est

$$A = \frac{\max\limits_{x_i, x_j \in S}(\delta(x_i, x_j))}{\min\limits_{x_k, x_l \in S}(\delta(x_k, x_l))}.$$

1.5.2 Dimension intrinsèque d'un espace métrique

Nous utilisons le terme *dimension intrinsèque* par opposition à la *dimension apparente* des données. Un jeu de données peut être constitué de vecteurs numériques avec un grand nombre de composantes et donc avoir une dimension apparente élevée. Si ces vecteurs correspondent à des points situées sur une surface de faible dimension plongée dans un espace de grande dimension, la dimension intrinsèque du jeu de données sera faible. La mesure de la dimension intrinsèque d'un jeu de données que nous utiliserons dans cette thèse est appelée la *dimension doublante*.

Définition 46 (Dimension de grille)
Soit (S, δ) un espace métrique. $B_u(r)$ désigne la boule de rayon r centrée sur le point $u \in S$. L'expansion de (S, δ) est notée $c_g = \max\limits_{u \in S, r \in \mathbb{R}^+} \frac{|B_u(2r)|}{|B_u(r)|}$. La dimension de grille de (S, δ), notée d_{grid}, est définie par $d_{grid} = \log_2 c_g$.

Intuitivement, la dimension de grille caractérise le nombre maximum de nouveaux points atteints quand le rayon d'exploration autour d'un point de l'espace double. Par exemple, si nous prenons tous les points qui composent une grille en D dimensions et la norme Euclidienne, alors $|B_u(r)| \sim r^D$ et la dimension de grille est égale à D.

Définition 47 (Dimension doublante [93])
Soit (S, δ) un espace métrique. Soit $p \in S$. Soit $C_{p,r}$ le nombre minimal de boules centrées en des points $\{u_i\}$ de S et de rayon $r/2$ telles que $B_p(r) \subseteq \bigcup\limits_{i=1}^{C_{p,r}} B_{u_i}(\frac{r}{2})$. La croissance doublante *c_{dd} associée à (S, δ) est $c_{dd} = \max\limits_{p \in S, r \in \mathbb{R}^+} \{C_{p,r}\}$.*

La dimension doublante *de* (S, δ)*, notée dd, est définie par*

$$dd = \log_2 c_{dd}. \tag{1.12}$$

Dans la thèse, nous considérons une version légèrement différente : $c_{dd} = \max_{p \in S, r=2^i} \{C_{p,r}\}$. La figure 1.30 illustre la couverture de boules de rayon r centrées sur des points de S par des boules de rayon $r/2$. La valeur maximum pour c_{dd} est 2, d'où une dimension doublante égale à 1. Seuls des algorithmes d'approximation existent pour le calcul de la dimension doublante [72, 79].

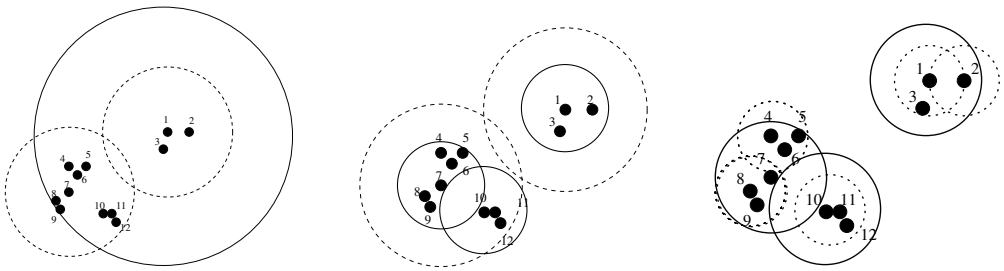


FIG. 1.30: Illustration de la dimension doublante d'un ensemble de points en 2D (c_d = 2, dd = 1). a) Couverture d'une boule de rayon r par deux boules de rayon $\frac{r}{2}$. b) Couverture des boules de rayon $\frac{r}{2}$ par des boules de rayon $\frac{r}{4}$ (deux au maximum). c) Couverture des boules de rayon $\frac{r}{4}$ par des boules de rayon $\frac{r}{8}$ (deux au maximum).

1.5.3 Echantillonage de l'espace métrique

Comme nous l'avons vu dans la section 1.4, les techniques de MDS qui apportent le meilleur compromis entre temps d'exécution et qualité de plongement ont recours à l'échantillonnage hiérarchique des sommets [60, 73, 70].

Nous utiliserons un échantillonage de l'espace métrique constituée d'une hiérarchie d'échantillons appelée *hiérarchie des centres discrets* [61]. A partir de cette hiérarchie d'échantillons, un *arbre d'échantillonage* T dont chaque niveau correspond à un niveau de la hiérarchie des centres discrets peut être construit.

Nous utilisons les définitions d'ensembles $k-$dominants et de $k-$séparation dans un graphe valué (cf. définitions 36 et 38) que nous étendons aux espaces métriques pour définir une hiérarchie des centres discrets.

Définition 48 (Hiérarchie des centres discrets)
Soit (S, δ) *un espace métrique. Soit* $S_h \subseteq S_{h-1} \subseteq \ldots \subseteq S_0 = S$ *une séquence de sous-ensembles de* S*. L'ensemble* $H_S = \{S_i\}_{0 \leq i \leq h}$ *est appelée* hiérarchie des centres discrets *de* (S, δ) *si et seulement si les deux propriétés suivantes sont vérifiées :*

- *couverture :* S_i *est une* 2^i*-domination de* S_{i-1}*.*
- *séparation :* S_i *est une* 2^i*-séparation de* S*.*

Un centre $u \in S_i$ *couvre* un point $v \in S_{i-1}$ si $\delta(u, v) \leq 2^i$. Un point $v \in S_{i-1}$ peut être couvert par plusieurs centres de S_i.

La définition 48 implique le lemme suivant, qui sera largement utilisé dans la suite de nos preuves.

Lemme 1.2 (S_i est une $2^{i+1}-$domination de S)
Soit $H_S = \{S_i\}_{0 \leq i \leq h}$ une hiérarchie des centres discrets de l'espace métrique (S, δ). S_i est une $2^{i+1}-$domination de S.

A chaque élément $u \in S$ on associe le niveau ℓ_u correspondant au niveau i d'indice maximal tel que $u \in S_i$ et donc $u \notin S_{i+1}$. Le niveau le plus élevé, h, contient au moins un élément. Par définition, S_h couvre S à distance 2^h donc $\forall u \in S, \forall v \in S_h, \delta(u, v) \leq 2^h$. Ceci implique que $\lceil \log A \rceil - 1 \leq h \leq \lceil \log A \rceil$ où A est l''aspect ratio" de S. Comme S_h est également une 2^h-séparation de S, il n'est composé que d'un seul élément.

A partir de la hiérarchie des centres discrets, on peut construire un arbre d'échantillonage T qui permet de représenter la hiérarchie des centres discrets (cf. figure 1.31).

Définition 49 (Arbre d'échantillonage T)
Soit $H_S = \{S_i\}_{0 \leq i \leq h}$ une hiérarchie des centres discrets de (S, δ). L'arbre d'échantillonage de (S, δ), noté T, est tel que $\forall u \in S_0$, $V(T)$ contient $1 + \ell_u$ copies de u. $\forall u \in V(T), \forall i \leq \ell_u \leq h$, le sommet u stocke :
- *$P(u) \in \{v \in S_{i+1} | \delta(u, v) \leq 2^{i+1}\}$, un* unique parent *de u choisi parmi l'ensemble des sommets de niveau $i+1$ qui couvrent u à distance 2^{i+1}.*
- *$C_{i-1}(u) = \{v \in S_{i-1} | P(v) = u\}$, l'ensemble des sommets couverts par u au niveau $i-1$. On appelle cet ensemble les* enfants de u au niveau $i-1$.

Définition 50 (Etiquetage des sommets de T)
Soit T un arbre d'échantillonage de (S, δ). On définit la fonction d'étiquetage de T, notée $Id : V(T) \rightarrow S_0$. Soit $u \in V(T)$:
- *$Id(u) \in \bigcup_{v \in C_{i-1}(u)} Id(v)$, $\forall i \geq 1$,*
- *$Id(u) = u$ si $i = 0$.*

On note $P_i(p^{(j)})$ l'ancêtre de $p \in S_j, j < i$ au niveau i. Le sommet u de niveau i est noté $u^{(i)}$. Pour chaque $i \leq \ell_u$, on définit l'arbre T_u^i enraciné sur la copie de u au niveau i. Quand $u \in S_h$, $T = T_u^h$ désigne l'arbre associé à l'espace métrique (S, δ).

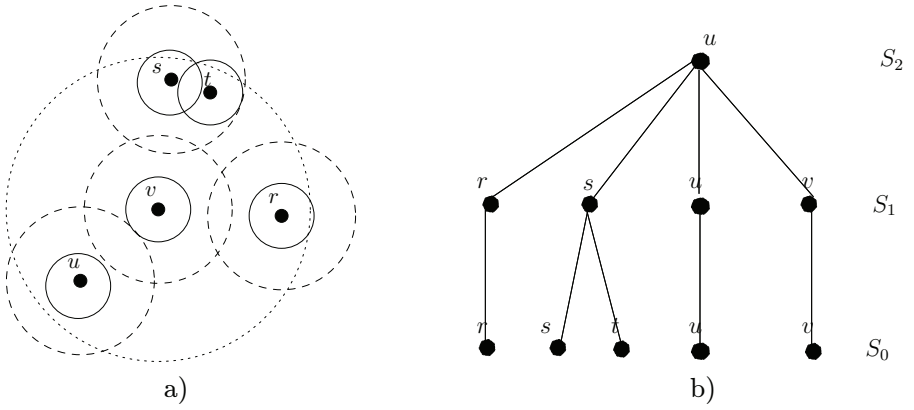


FIG. 1.31: Exemple d'arbre d'échantillonage T. a) Données S composées de 5 points dans le plan. b) Un arbre d'échantillonage T possible pour S avec 3 niveaux.

1.5.3.1 Construction de T

L'arbre d'échantillonage T respectant les propriétés de couverture et de séparation peut être obtenu en utilisant une des structures de données présentées dans cette section.

"Deformable spanner" L'algorithme présenté dans [61], permet de construire un $1+\epsilon$-spanner (cf. définition 35) d'un ensemble de points de l'espace Euclidien de dimension d. Les auteurs montrent également que leur structure de données permet de résoudre efficacement les problèmes suivants :

- trouver $u, v \in S$ tels que $\min_{u,v \in S} \delta(u, v)$,
- trouver les k plus proches voisins de chaque sommet,
- donner une $8-$approximation du problème des k-centres.

Le "Deformable spanner" permet également d'ajouter et de supprimer des points de manière incrémentale.

Le "Deformable Spanner" est composé d'un arbre d'échantillonage T qui respecte les propriétés de séparation et de couverture présentées dans la section 1.5.3. Pour compléter la construction du "Deformable Spanner", les paires de sommets qui appartiennent à l'ensemble S_i, entre lesquels la distance est inférieure à $c.2^i$ sont reliés par une arête. La valeur $c = 4 + 16/\epsilon$ permet de contrôler le facteur d'étirement du "Deformable Spanner".

"Cover Tree" La structure de données proposée dans [21] est également utilisée pour le calcul exact ou approximatif des plus proches voisins dans un espace métrique.

Le "Cover Tree" est un arbre qui respecte les mêmes propriétés que l'arbre T. Les auteurs présentent notamment un codage de l'arbre qui a une complexité en espace linéaire.

Caractéristiques	"Deformable Spanner" [61]	"Cover Tree" [21]
Espace (arêtes)	$O(n2^{O(dd)} \log A)$	$O(n)$
Temps construction	$O(n2^{O(dd)} \log A)$	$O(c_g^6 n \log n)$
Temps de recherche	$O(2^{O(dd)} \log A)$	$c_g \log A$
Spanner	oui	non

TAB. 1.2: Caractéristiques des algorithmes permettant de construire un arbre d'échantillonage.

Le Tableau 1.2 résume les caractéristiques importantes des deux algorithmes permettant de calculer l'arbre T. L'algorithme le moins coûteux en espace est le "Cover Tree", cependant, si l'utilisation de distance approximatives dans l'espace métrique est suffisante, le "Deformable Spanner" peut être une bonne alternative au stockage de la matrice de distances.

Lemme 1.3 (Temps de construction de l'arbre d'échantillonage [61])
Soit (S, δ) un espace métrique de dimension doublante dd, d'"aspect ratio" A et contenant n éléments. L'algorithme "Deformable Spanner" permet de construire l'arbre d'échantillonage T en temps $O(n2^{O(dd)} \log A)$.

Chapitre 2

Structuration de document vidéo par fragmentation de graphe

Dans ce chapitre, nous présentons une méthode permettant d'identifier des scènes composées de plans vidéo similaires dans un document vidéo [43, 42, 41]. Cette méthode se base sur la représentation des similarités entre les plans d'un document vidéo par un graphe valué que nous appellons *graphe vidéo*. Les graphes sont adaptés à la représentation d'un document vidéo sous forme d'entités associées aux plans vidéo et munies d'attributs (image-clé, durée du plan, valeur de descripteur) et de relations entre plans vidéo également munies d'attributs (distance entre plans dans le document, mesure de similarité).

Un filtrage des arêtes du graphe est appliqué pour supprimer les arêtes correspondant aux faibles similarités entre les plans. Ce filtrage induit une structure dans le graphe qui reflète la présence de scènes dans le document vidéo associé. La fragmentation du graphe filtré permet d'identifier des groupes de plans au contenu homogène au sens du descripteur utilisé. La technique de fragmentation proposée est générique dans la mesure où elle se base sur un paramètre intrinsèque du graphe.

L'analyse des caractéristiques de bas niveau d'un document vidéo peut permettre de détecter des évènements sémantiques de haut niveau. Par exemple, la détection de séquences contenant des fréquences élevées dans le signal audio permet d'identifier les buts dans les programmes de football [88]. L'analyse des couleurs de l'image peut fournir une indication sur la présence de visages [55].

Aussi, dans le contexte de l'analyse des documents vidéo, le groupement des plans vidéo partageant un contenu similaire au sens d'une ou plusieurs caractéristiques de bas niveau, permet la reconstruction :

- de *scènes* quand les plans groupés sont également proches dans le temps [176],
- d'*hyper-scènes* [122] quand ces plans similaires sont distants dans le temps.

Les hyper-scènes permettent d'identifier des séquences récurrentes dans un document vidéo (présentateur, générique, publicité, etc).

Dans [176], les auteurs utilisent un algorithme de fragmentation agglomératif ("complete link") pour former des scènes. La mesure de similarité tient compte de la couleur et de la luminance des plans mais également du temps séparant les plans. Ainsi une paire de plans similaires mais trop éloignés dans le temps ne peut pas être groupée dans le même fragment. Un graphe de transition de scène est ensuite construit en reliant deux fragments par une arête si deux plans consécutifs dans le document vidéo appartiennent à ces fragments. Les arêtes séparatrices du graphe de transition de scène délimitent des scènes ou "story units".

Notre approche pour le groupement de plans en scènes ou en hyper-scènes est basée

sur la fragmentation naturelle du graphe associé à un document vidéo. La fragmentation naturelle de graphe ("natural graph clustering") est un sous-ensemble des techniques de fragmentation de graphes dans lesquelles une fragmentation est optimale si elle représente les "groupes naturels" inclus dans le graphe. La notion de groupe naturel dépend de la nature des données codées par le graphe. Par exemple, les différents domaines de recherche traités dans une communauté scientifique peuvent être identifiés par une fragmentation du graphe de co-citation des publications scientifiques [82].

Il existe différentes caractérisations des clusters naturels dans un graphe G :

- Le nombre de chemins de longueur supérieure à 1 entre deux sommets u et v est grand si u et v appartiennent à un même fragment dense et petit si u et v appartiennent à des fragments différents. Cette propriété est utilisée dans [149].
- Les arêtes séparant des fragments différents sont susceptibles d'apparaitre dans de nombreux plus courts chemins du graphe G. Cette caractéristique s'appelle la "betweeness" d'une arête et est utilisée pour la fragmentation de graphe dans [121].
- Le coefficient de "clustering" [168] d'un sommet v mesure le degré d'appartenance d'un sommet à un fragment dense. Un coefficient de "clustering" égal à 1 indique l'appartenance de v à une clique. Dans [7], cette mesure est étendue aux arêtes du graphe afin de déterminer les arêtes interconnectant des fragments denses. La suppression de ces arêtes isole les fragments.

Notre approche de la fragmentation du graphe en clusters naturels repose sur le calcul d'un paramètre combinatoire intrinsèque associé aux sommets d'un graphe orienté sans cycle, le nombre de Strahler. La valeur du nombre de Strahler d'un sommet du DAG permet de refléter partiellement sa structure (profondeur de l'arbre couvrant, degré de ramifications). Notre méthode attribue successivement à chacun des sommets du graphe à fragmenter le rôle de racine du DAG et son nombre de Strahler associé est calculé. Si les sommets appartiennent à un fragment dense alors la méthode de retournement d'arête utilisée pour induire chaque DAG conduit à former des DAG avec une structure similaire. Les groupes de DAG de structure similaire sont ensuite identifiés par le calcul du nombre de Strahler sur ces DAG..

Dans la première section, nous présentons la méthode d'extraction de la signature couleur d'un plan vidéo, utilisée pour comparer les plans vidéo selon leur similarité. Nous présentons ensuite la construction du graphe associé à un document vidéo, ainsi que sa fragmentation en hyper-scènes. Enfin, nous évaluons la méthode en comparant la fragmentation produite et les scènes associées avec une indexation de référence

2.1 Indexation des scènes d'un document vidéo basée sur la couleur

Le groupement des plans vidéo en scènes consiste à reconstruire la sémantique d'un document vidéo à partir d'une segmentation temporelle de bas niveau et de sa description : les plans vidéo et leurs descripteurs.

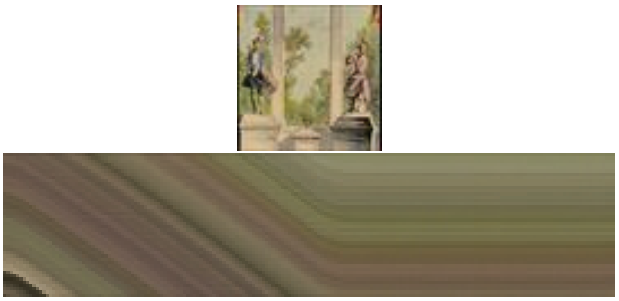
Par définition (cf. section 1.1.2), les plans qui partagent une valeur de descripteur similaire appartiennent à la même scène. Les descripteurs de couleur et de texture issus de la norme MPEG7, section 3, [34] peuvent être utilisés mais on utilisera ici un descripteur dédié au plan vidéo : la signature couleur spatio-temporelle du plan vidéo [19].

2.1.1 Signature spatio-temporelle des plans vidéo

Plusieurs méthodes ont été proposées pour caractériser le contenu couleur des plans vidéo [100, 176]. Ces méthodes incluent notament l'utilisation d'histogrammes couleur comme dans le descripteur ISO/MPEG-7 "GOP/GOF Descriptor" [15]. Il est également envisageable d'utiliser un ensemble de descripteurs issus d'un ensemble de "frames" issues d'un plan vidéo pour en caractériser le contenu. Dans [42], nous avons comparé l'efficacité de cette approche avec les descripteurs "Dominant Color Descriptor" et "Color Layout Descriptor" et l'utilisation de la signature spatio-temporelle d'un plan vidéo. Notre conclusion est que la signature spatio-temporelle basée sur la projection "X-ray" représente un bon compromis entre performance et temps de calcul des mesures de dissimilarités entre descripteurs.

La signature couleur produite par la projection "X-ray" (cf. Définition 2) d'une "frame" conserve la distribution spatiale de l'intensité de l'image dans la direction orthogonale à l'axe de projection. C'est une propriété intéressante de cette transformée par rapport aux histogrammes.

La figure 2.1 illustre la conservation de la distribution spatiale des couleurs par la projection "X-Ray". Des projections de "frames" DC sont cumulées dans le temps et composent l'image présentée dans la figure 2.1. La figure 1.b) correspond à un plan vidéo montrant la bouche d'un interlocuteur en plan rapproché. La figure 1.a) représente un traveling vertical sur une peinture. D'une part, on peut voir que la projection "X-Ray" résume le contenu couleur du plan vidéo entier (cf. la figure 2.1.a)). D'autre part, elle peut exprimer la composante verticale du mouvement présent dans le plan vidéo. Ainsi, dans la figure 2.1.a), la pente des lignes change avec le traveling (vertical) de la caméra.



a)

b)

FIG. 2.1: Image-clé en résolution DC et projections "X-Ray" cumulées. a) Plan 17, "La joueuse de Tympanon", SFRS©. b) Plan 18, "La joueuse de Tympanon", SFRS©.

2.1.2 Mesure de dissimilarité de la signature spatio-temporelle

Dans [100], une approche basée sur le codage a été proposée pour mesurer la dissimilarité entre les plans vidéo basée sur le quantification vectorielle [64].

En compression de données, la quantification vectorielle est utilisée comme une technique de compression avec pertes. Le principe consiste à remplacer une valeur dans un espace métrique par un code ou clé. Le code peut être une valeur d'un espace vectoriel de plus petite dimension, obtenu par projection mais il peut aussi s'agir d'une clé issue de l'espace vectoriel original mais utilisée pour coder plusieurs vecteurs distincts à la fois. Dans les deux cas, la compression des données est obtenue par perte d'information - projection ou approximation de la valeur à coder- et l'opération de décodage n'est pas réversible, d'où le terme de compression avec perte.

Définition 51 (Dictionnaire)
Soit $E \subset \mathbb{R}^d$ un espace vectoriel, soit $F \subset E$. On appelle fonction de codage, l'application $f_E : E \rightarrow F$, qui associe à chaque élément de E une clé $k \in F$, i.e. $F = f_E(E)$. On appelle dictionnaire associé à E, l'ensemble F.

Le choix d'un dictionnaire doit être suffisament judicieux pour que l'écart entre l'élément codé et sa clé soit minimisé. Cet écart est mesuré par la distance entre deux éléments de l'espace E. Soient $\overrightarrow{u}, \overrightarrow{v}, \in E$ on note $||\overrightarrow{u}, \overrightarrow{v}||$, la distance entre $\overrightarrow{u}$ et $\overrightarrow{v}$ dans E.

Dans [139], chaque "frame" est segmentée en blocs rectangulaires disjoints de taille $M * N$. Puis, pour un plan vidéo particulier, chaque bloc issu de chaque "frame" est considéré comme un vecteur dans l'espace $\mathbb{R}^{3MN}$. Un dictionnaire est créé pour chaque plan à partir des blocs issus de ce plan. La construction du dictionnaire utilise un algorithme de Quantification Vectorielle (QV). Une fois que chaque plan dispose de son dictionnaire, on peut mesurer la similarité entre le contenu de deux plans vidéo en comparant la distorsion de codage engendrée par l'utilisation du dictionnaire associé à l'un des deux plans vidéo pour coder les valeurs de l'autre plan vidéo. Si deux plans vidéo ont un contenu identique, alors ils sont composés des mêmes blocs, qui devraient être codés par le même dictionnaire. Plus le contenu de deux plans vidéo diffère, plus les dictionnaires associés diffèreront également et la distorsion de codage engendrée sera grande.

Définition 52 (Distorsion de codage propre)
Soit l'ensemble $E \subset \mathbb{R}^d$, soit f_E la fonction de codage associée à E. La distorsion de codage propre engendrée par le codage des éléments de E par sa fonction de codage f_E est définie par :

$$D_{f_E}(E) = \sum_{\overrightarrow{u} \in E} ||\overrightarrow{u}, f_E(\overrightarrow{u})||^2 \tag{2.1}$$

Définition 53 (Distorsion de codage croisée)
Soient les ensembles $E \subset \mathbb{R}^d$ et $G \subset \mathbb{R}^d$, soit f_E la fonction de codage associée à E. La distorsion de codage croisée engendrée par le codage des éléments de G par la fonction de codage f_E associée à l'ensemble E est définie par :

$$D_{f_E}(G) = \sum_{\overrightarrow{u} \in G} ||\overrightarrow{u}, f_E(\overrightarrow{u})||^2 \tag{2.2}$$

Sur la base de la distorsion de codage, une mesure de dissimilarité entre ensembles de vecteurs est définie [19].

Définition 54 (Dissimilarité basée sur la distorsion de codage)
Soient les ensembles $E \subset \mathbb{R}^d$ et $G \subset \mathbb{R}^d$, soit f_E la fonction de codage associée à E et f_G la fonction de codage associée à G. La mesure de dissimilarité entre E et G , notée $\delta'(E,G)$, basée sur la distorsion de codage, est définie par

$$\delta'(E,G) = |D_{f_G}(E) - D_{f_E}(E)| + |D_{f_E}(G) - D_{f_G}(G)| \tag{2.3}$$

Soient $F, G, H \subset \mathbb{R}^d$. On vérifie facilement que cette mesure respecte deux des trois propriétés d'une distance :
- $\delta'(E,E) = 0$
- $\delta'(E,G) = \delta'(G,E)$

Concernant l'inégalité triangulaire, cette propriété est satisfaite dans le cas où les distorsions de codage sont toutes identiques : $D_{f_F}(F) = D_{f_G}(G) = D_{f_H}(H)$. L'inégalité triangulaire n'est donc pas garantie en général et δ' est une pseudo-distance.

On applique cette technique à l'ensemble des "bins" issus des projections "X-Ray" des "frames" de chaque plan vidéo en calculant le dictionnaire qui détermine sa fonction de codage.

Définition 55 (Signature spatio-temporelle)
Soit un plan vidéo $P = I_m, .., I_n$ constitué de $n-m+1$ "frames" de hauteur H et $Bins_P = \{\mathcal{X}_y^{I_i}(x) \in \mathbb{N}^3 | \forall x \in [1,H], \forall i \in [m,n]\}$ l'ensemble des "bins" calculés à partir des "frames" de P. On appelle signature spatio-temporelle du plan P, notée S_P, le dictionnaire calculé pour l'ensemble $Bins_P$.

Les éléments de la signature spatio-temporelle d'un plan vidéo sont donc des valeurs de couleur exprimés dans l'espace YUV. Soit $\overrightarrow{u} \in Bins_P$, la fonction de codage de $\overrightarrow{u}$ par un élément de son dictionnaire S_P est définie par :

$$f_{Bins_P}(\overrightarrow{u}) = \min_{\overrightarrow{v} \in S_P} ||\overrightarrow{u}, \overrightarrow{v}||. \tag{2.4}$$

Une technique fréquente pour le calcul de dictionnaires par quantification vectorielle est la méthode de Split-LBG [64], qui est une généralisation de la méthode de fragmentation par les K-means [104]. Cette méthode a été utilisée pour construire les signatures spatio-temporelles des plans vidéo car elle produit directement un dictionnaire compatible avec la formule 2.5. Le nombre maximum d'entrées dans le dictionnaire est également un paramètre de SplitLBG utile pour contrôler le niveau de distorsion de codage propre de chaque dictionnaire.

Définition 56 (Dissimilarité entre les plans)
Soient P_i, resp. P_j, deux plans vidéo, soient $Bins_{P_i}$, resp. $Bins_{P_j}$, l'ensemble des "bins" calculés d'après P_i et P_j et $f_{Bins_{P_i}}$, resp. $f_{Bins_{P_j}}$, les fonctions de codage associées. La dissimilarité entre P_i et P_j, notée $\delta(P_i,P_j)$, basée sur la signature spatio-temporelle entre plans est :

$$\delta(P_i,P_j) = \delta'(Bins_{P_i}, Bins_{P_j}) \tag{2.5}$$

2.1.3 Définition du "graphe vidéo"

La mesure de dissimilarité entre plans vidéo définie dans la section précédente permet de calculer une matrice D, de taille $N \times N$ dont les éléments sont $\delta_{ij} = \delta(P_i,P_j)$. D'après

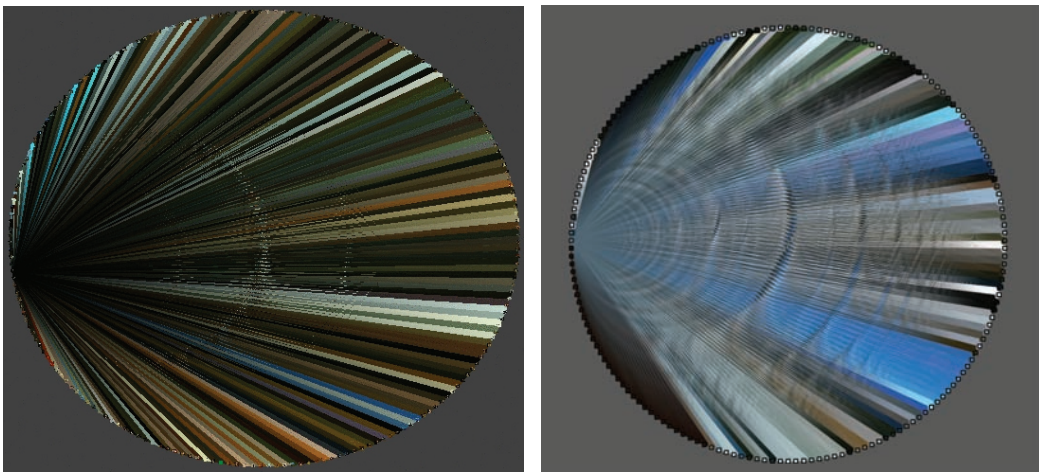
la définition de $\delta(P_i, P_j)$, la matrice est symétrique et contient des 0 sur la diagonale principale. On peut donc réduire D à la matrice triangulaire

$$D = (\delta_{ij})_{1 \leq i \leq N-1, i+1 \leq j \leq N}$$

où N désigne le nombre de plans dans le document vidéo traité.

Cette matrice peut être vue comme la matrice d'incidence d'un graphe complet orienté et valué. Un tel graphe sera utilisé dans la suite de ce chapitre comme une abstraction d'un document vidéo et sera désigné par le terme "graphe vidéo". En suivant l'ordre naturel des sommets (l'ordre des plans dans le document vidéo), on obtient un DAG. Soit une arête $e = (i, j)$, si $i < j$, alors e appartient au DAG et (j, i) n'appartient pas au DAG. Cet ordre conserve l'ordre chronologique des plans. Le sommet N n'ayant que des arêtes entrantes et aucune arête sortante, est un *puits* du DAG.

La figure 2.2 montre une représentation d'un graphe vidéo correspondant à deux documentaires. La couleur correspondant à la classe d'effectif maximum dans l'histogramme des couleurs de l'image-clé (la couleur dominante) est plaquée sur les arêtes du graphe ce qui donne un aperçu du contenu couleur global de l'ensemble du document vidéo.



a) b)

FIG. 2.2: Graphes vidéo. a) Graphe vidéo composé de $|V| = 200$ plans extraits de la séquence intitulée "La joueuse de Tympanon" SFRS©. b) Graphe vidéo composé de $|V| = 174$ plans extraits de la séquence intitulée "Aquaculture en Méditerranée" SFRS©.

2.2 Fragmentation de graphe basée sur la structure

L'objectif de la technique présentée dans cette section consiste à décomposer le graphe vidéo en fragments dont les sommets correspondent à des plans similaires. Ces fragments correspondent à des hyper-scènes du document vidéo représenté par le graphe vidéo. Ces hyper-scènes répondent au critère d'homogénéité de leur contenu couleur au sens du contenu couleur exprimé par leur descripteur "X-Ray". L'approche est celle de la visualisation d'information et il s'agit de permettre à l'utilisateur de manipuler la représentation d'un document vidéo sous forme d'un graphe pondéré.

Dans une première étape, on induit une structure de graphe en fonction de la valeur de dissimilarité utilisée pour valuer les arêtes du graphe vidéo. Cette approche consiste à filtrer les arêtes du graphe ayant une valuation élevée. Si on se réfère à la matrice de dissimilarité associée au document vidéo, cela signifie que la mesure de dissimilarité entre les deux plans reliés par une arête filtrée est élevée. La suppression de cette arête empêche les sommets reliés d'être voisins dans le graphe filtré.

Le filtrage des arêtes n'est pas suffisant pour isoler tous les fragments du graphe qui correspondent à des hyper-scènes valides. Parmi les composantes connexes produites par l'étape de filtrage, certaines correspondent à un mélange de plusieurs hyper-scènes. On propose donc une méthode de fragmentation qui permet de fragmenter ces composantes connexes en fragments composés exclusivement de plans ayant un contenu couleur similaire. L'algorithme tend à identifier des quasi-cliques dans chaque composante connexe. Les plans associés aux sommets des quasi-cliques se rappochent par leur contenu couleur et correspondent donc à la notion d'hyper-scènes du document vidéo que l'on souhaite indexer.

Notre hypothèse est que les sommets d'une même quasi-clique ont un voisinage similaire : les ensembles de sommets atteignables à $1, 2, ..., k$ sauts de chaque sommet d'une quasi-clique sont constitués d'approximativement les mêmes sommets. Nous représentons le voisinage d'un sommet par un arbre (resp. un DAG) enraciné en ce sommet et nous supposons que les arbres (resp. les DAG) enracinés en des sommets appartenant à la même quasi-clique auront une structure similaire, c'est à dire que leur aspect général, leur degré de ramification, ou leur profondeur, seront comparables. Pour mesurer cette similarité structurelle, nous utilisons un paramètre combinatoire appelé nombre de Strahler [146] qui s'applique aux arbres généraux et aux DAG et qui permet de caractériser une partie de la structure de ces graphes.

2.2.1 Filtrage statistique des arêtes

Le problème du filtrage des arêtes consiste à déterminer un seuil de dissimilarité au dessus duquel les arêtes dont la valuation excède ce seuil seront supprimées. On propose de déterminer ce seuil selon les caractéristiques de la fonction de distribution des valeurs de dissimilarités associée à un document vidéo.

Soit f_c l'histogramme de la valuation des arêtes du graphe vidéo $G = (V, E)$ et soit Q_{max} la valeur maximum de la valuation d'une arête.

La figure 2.3 montre la fonction de répartition des valeurs de dissimilarité des plans de deux documents vidéo ("La joueuse de Tympanon" et "Aquaculture en Méditerranée", SFRS©). On peut constater que les deux distributions sont similaires. On remarque également que les distributions atteignent un maximum. Soit $q_0 \in [0; Q_{max}]$ tel que $f_c(q_0)$ est maximal. Pour un graphe vidéo $G = (V, E)$ donné, on considérera le filtrage de l'ensemble d'arêtes $E' = \{e = (i, j) \in E | \delta_{ij} > q_0\}$. Le graphe vidéo privé des arêtes dont la valuation correspond à des plans vidéo peu similaires est $G' = (V, E - E')$.

Ce premier filtrage du graphe vidéo fournit une première fragmentation du graphe. La figure 2.4 montre une représentation du graphe G' associé à un document vidéo ("Chancre coloré du platane", SFRS©). On peut constater que quelques sommets isolés ainsi que des petits fragments ont été déconnectés du reste du graphe. Ces composantes connexes sont considérées comme un ensemble de scènes indépendantes.

L'étape de filtrage du graphe vidéo n'est pas suffisante pour permettre une décomposition complète en scènes couleur car des fragments composés de plans vidéo au contenu couleur hétérogène subsistent. On propose donc de fragmenter d'avantage ces composantes

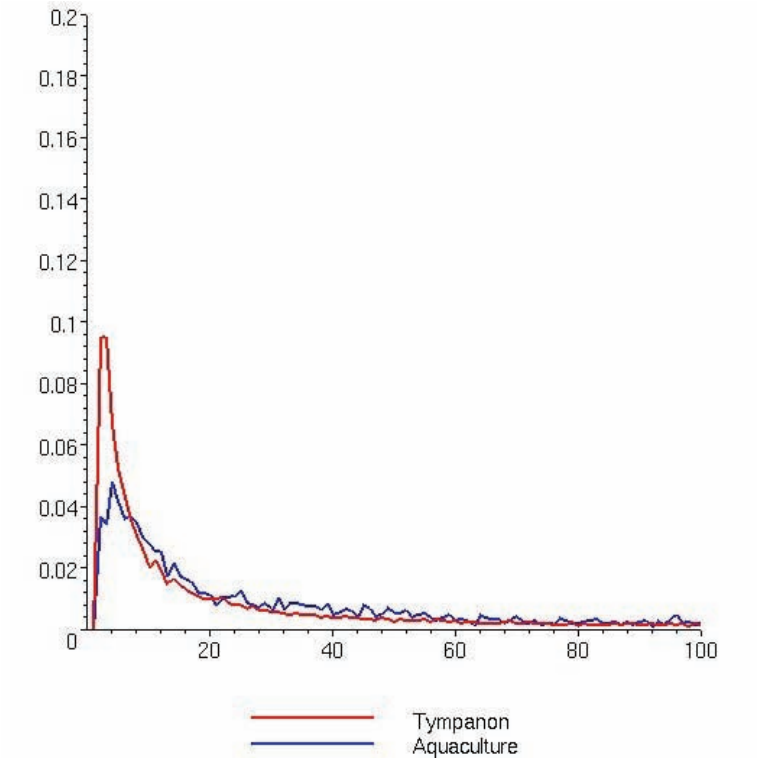


FIG. 2.3: Distribution des valeurs de dissimilarité des documents "La joueuse de Tympanon" et "Aquaculture en Méditerranée" SFRS©.

connexes selon une méthode utilisant les caractéristiques structurelles de ces graphes. La structure des sous-graphes est induite par l'étape de filtrage des arêtes de poids élevé. Il y a donc une relation entre la structure du graphe et la similarité des plans des sommets qui le composent.

2.2.2 Fragmentation basée sur la structure

Le filtrage précédent du graphe vidéo ayant supprimé les arêtes reliant les sommets dissimilaires, il est naturel de penser que les fragments recherchés sont composés de sommets fortement connectés entre eux et faiblement connectés avec les autres sommets. Étant donné un sous-graphe issu de la décomposition en composantes connexes de G', on souhaite fragmenter les quasi-cliques qu'il contient.

La recherche d'une $1-$ *clique* maximale (cf. Définition 27) dans un graphe est un problème $NP-$difficile. Dans cette section, nous proposons une heuristique pour la fragmentation d'un graphe en quasi-cliques les plus denses possibles.

L'approche que nous présentons se base sur les deux étapes suivantes :

1. Calculer une valuation des sommets selon un paramètre combinatoire adapté,
2. Grouper les sommets possédant une valuation similaire en classes d'équivalence, et former un fragment par classe d'équivalence.

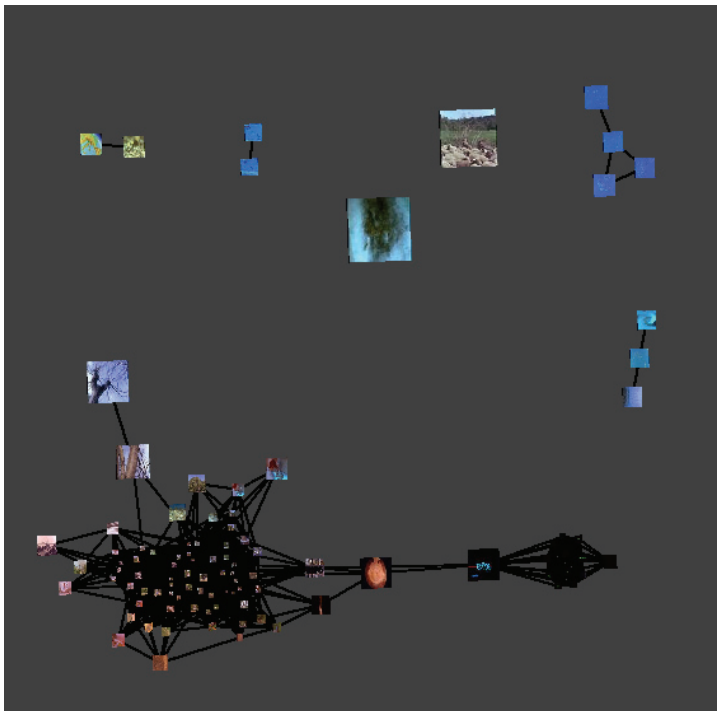


FIG. 2.4: Fragments après l'étape de filtrage du graphe associé à la séquence "Chancre coloré du platane" SFRS©.

2.2.2.1 Valuation des sommets par les nombres de Strahler

Nombre de Strahler Le nombre de Strahler d'un arbre binaire, noté σ_B, a été introduit dans une étude sur la structure morphologique des fleuves [146]. Une valeur entière est calculée pour chaque sommet de l'arbre binaire. Cette valeur fournit une information quantitative sur la complexité du sous-arbre correspondant. Un algorithme récursif permet de calculer cette valeur en chaque sommet.

Définition 57 (Nombre de Strahler d'un arbre binaire)
Soit $T = (V, E)$ un arbre binaire, soit un sommet $v \in V$.
- *Si v est une feuille de T, alors $\sigma_B(v) = 1$.*
- *Sinon, v a deux sous-arbres enracinés en v_1 et v_2 et :*

$$\sigma_B(v) = \begin{cases} max(\sigma_B(v_1), \sigma_B(v_2)) \text{ si } \sigma_B(v_1) \neq \sigma_B(v_2) \\ \sigma_B(v_1) + 1 \text{ sinon} \end{cases}$$

Une interprétation du nombre de Strahler des arbres binaires a été proposée par Ershov [52] qui a prouvé que le nombre de Strahler incrémenté de un est exactement le nombre de registres minimum requis pour calculer une expression arithmétique donnée sous la forme d'un arbre d'analyse syntaxique.

Le nombre de Strahler d'un arbre général, noté σ_T est défini dans [5].

Définition 58 (Nombre de Strahler d'un arbre général)
Soit $T = (V, E)$ un arbre général, soit un sommet $v \in V$.
- *Si v est une feuille alors $\sigma_T(v) = 1$*

– *Sinon v a $k+1$ sous-arbres enracinés en $v_0, v_1, ..., v_k$ et classés par ordre décroissant du nombre de Strahler à leur racine et*

$$\sigma_T(v) = \max_{0 \leq i \leq k} (\sigma_T(v_i) + i)$$

L'extension aux DAG est directe. Soit $G = (V, E)$ un DAG. Chaque DAG a au moins une racine qui est un sommet sans arête entrante. Chaque DAG a au moins un puits. Le nombre de Strahler σ_D sur les DAG suit la définition précédente dans laquelle le terme feuille est remplacé par puits et sous-arbre est remplacé par sous-DAG.

Définition 59 (Nombre de Strahler d'un DAG)
Soit $T = (V, E)$ un DAG, soit un sommet $v \in V$.
– *Si v est un puits alors $\sigma_D(v) = 1$*
– *Sinon v a $k+1$ sous-DAG enracinés en $v_0, v_1, ..., v_k$ et classés par ordre décroissant du nombre de Strahler à leur racine et*

$$\sigma_D(v) = \max_{0 \leq i \leq k} (\sigma_D(v_i) + i)$$

La figure 2.5 présente les valeurs de Strahler, σ_D, associées aux sommets d'un DAG.

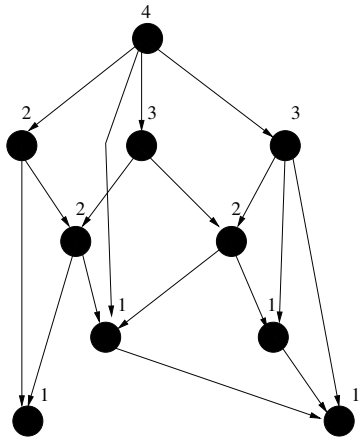


FIG. 2.5: DAG avec les sommets valués avec le nombre de Strahler associé (cf. Définition 59).

Famille de DAG Nous présentons ci-dessous la construction de la famille de DAG issus de chacun des sommets du graphe à fragmenter. Soit G, le graphe à fragmenter. Par construction, G est lui-même un DAG. La famille de DAG que l'on construit représente un sous-ensemble de tous les DAG composés du même ensemble de sommets et du même ensemble d'arêtes, mais orientées différemment. La famille de DAG est construite en transformant la racine du DAG précédent en puits. Cette transformation s'obtient en retournant les arêtes sortantes de la racine précédente.

Pour un DAG particulier, la famille associée est notée $\mathcal{F} = \{DAG(i)\}_{1 \leq i \leq |V|}$ et DAG(1) est le graphe initial. Soit $DAG(i) = (V, E_i)$ avec V l'ensemble de sommets et E_i l'ensemble d'arêtes orientées. Le graphe $DAG(i+1) = (V, E_{i+1})$, est défini par :

$$E_{i+1} = \{(j,k) \in E_i | j \neq i\} \bigcup \{(k,i) | (i,k) \in E_i\} \tag{2.6}$$

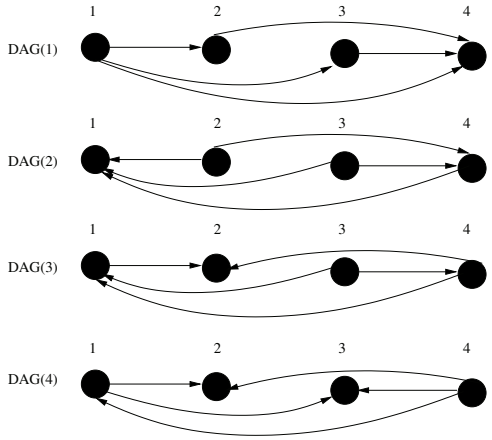


FIG. 2.6: Famille de DAG.

La figure 2.6 montre une famille de 4 DAG. On remarque que $DAG(n+1) = DAG(1)$. Dans chaque $DAG(i)$, le sommet i est la racine et le sommet $i-1$ est un puits du DAG.

Soit G' un sous-graphe du graphe vidéo, produit par l'étape de filtrage. On associe à chaque sommet i de G', le nombre de Strahler calculé sur le graphe $DAG(i)$ qui est le DAG de la famille dont la racine est le sommet i. On désigne cette valeur par $\sigma_D(i)$. Ainsi, on obtient un graphe dont les sommets sont valués. Les DAG de la famille présentée dans la figure 2.6 conduisent aux valeurs suivantes

$$\sigma_D(1) = 3, \sigma_D(2) = 2, \sigma_D(3) = 2, \sigma_D(4) = 3$$

On considère que dans ce graphe, les sommets 1 et 4 sont dans la même situation car ils partagent la même valeur de Strahler.

2.2.2.2 Fragmentation par convolution

Pour permettre de grouper en classes d'équivalence les sommets dont la valeur de Strahler, σ_D est proche, on utilise une technique permettant d'effectuer cette tâche de manière interactive. Cette technique a été proposée par Auber et al. dans [8], il s'agit de la fragmentation par convolution.

Pour chaque composante connexe produite par le processus de filtrage, la famille de DAG associée est construite et la valeur du nombre de Strahler est calculée à la racine de chaque DAG. Un histogramme des valeurs de Strahler est construit et les minima locaux de l'histogramme définissent les frontières des classes d'équivalence de sommets partageant la même valeur de Strahler. Si le nombre et la répartition des minima locaux de l'histogramme sont satisfaisants, alors les sous-graphes induits par les sommets de chaque classe d'équivalence sont construits et le graphe quotient correspondant est affiché à l'utilisateur. Si le nombre de classes d'équivalence est trop important, ou que l'utilisateur juge que des classes voisines devraient être fusionnées, il est possible de "lisser" l'histogramme en lui appliquant une opération de convolution avec une autre fonction qui peut être par exemple, une Gaussienne ou une fonction triangulaire.

Exemple : La figure 2.7 montre la convolution d'un exemple d'histogramme $x[n]$ avec une fonction de convolution triangulaire $h[n]$. Intuitivement, quand le centre de la fonction triangulaire coincide avec un minimum local du signal, le produit de convolution à cet

endroit correspond à une "moyenne pondérée" des échantillons de signal couverts par le noyau. Ainsi, plus le noyau est large, plus le signal $x[n]$ est lissé.

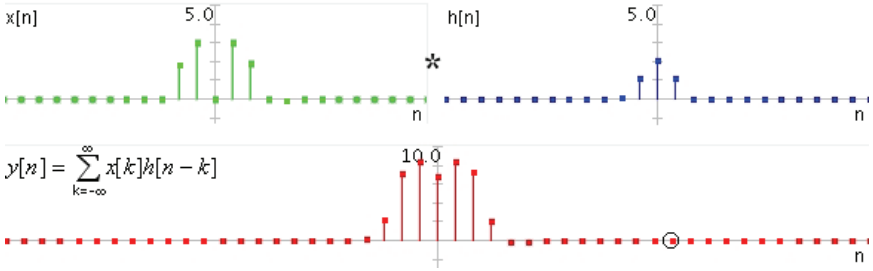
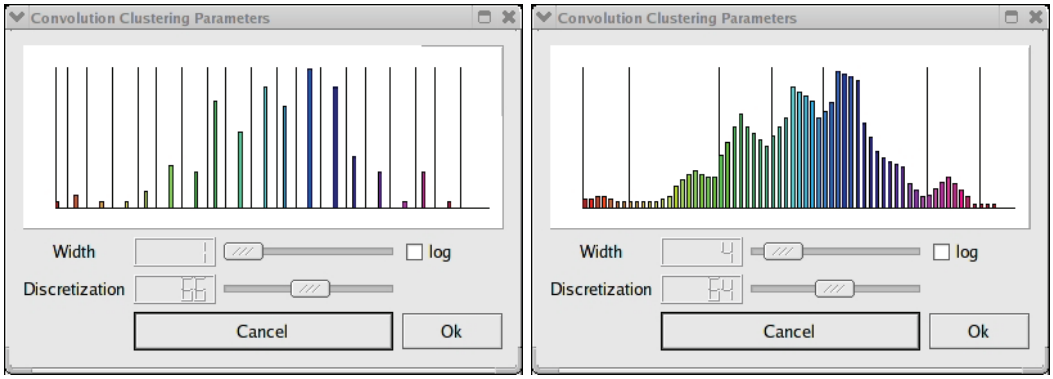


FIG. 2.7: Convolution du signal discret $x[n]$ par le noyau de convolution $h[n]$. Le signal $y[n]$ est obtenu, on remarque que la valeur de l'abscisse 0 de $x[n]$ n'est plus nulle. (Illustration produite avec l'applet http ://www.jhu.edu/∼signals/convolve/index.html).

La figure 2.8 illustre l'application d'une convolution sur le support d'un histogramme dont on souhaite réduire le nombre de minima locaux. La figure 2.8 b) montre l'effet de l'élargissement du noyau sur l'histogramme.



a) b)

FIG. 2.8: Application d'une convolution sur le support d'un histogramme. a) Histogramme initial. b) Histogramme après convolution. Le paramètre "width" qui contrôle la largeur du noyau de convolution a été augmenté.

Cet outil évite l'ajout ou la suppression individuelle des frontières entre classes d'équivalence. Cette technique est détaillée dans [8].

La fragmentation est effectuée de manière interactive, dans le logiciel Tulip [6]. Les fragments produits sont représentés sous la forme d'un graphe quotient dont chaque sommet correspond à un fragment. La qualité des fragments peut être évaluée par l'examen des images-clé plaquées sur chaque sommet du graphe vidéo.

Cette technique a été utilisée pour fragmenter les documentaires dont les caractéristiques sont décrites dans le Tableau 2.1. Dans ce tableau, le nombre de sommets $|V|$ du graphe correspond au nombre de plans. Le nombre d'arêtes $|E|$ correspond au nombre d'arêtes du graphe vidéo avant l'étape de filtrage.

La figure 2.9 montre une vue globale du graphe quotient associé aux 17 fragments issus de la séquence "La joueuse de Tympanon" SFRS©. La figure 2.11 montre une vue détaillée sur un fragment de la séquence "Chancre coloré du platane" SFRS ©. La figure 2.12 montre une vue détaillée sur un fragment de la séquence "Aquaculture en Méditerranée" SFRS ©.

Nom de la séquence	Durée (minutes)	Taille (plans)	$\|V\|$	$\|E\|$
De l'arbre à l'ouvrage, SFRS©	52	321	321	51360
La joueuse de Tympanon, SFRS©	26	200	200	19900
Aquaculture en Méditerranée, SFRS©	13	87	87	3741
Chancre coloré du platane, SFRS©	16	98	98	4753

TAB. 2.1: Caractéristiques des séquences vidéo traitées.

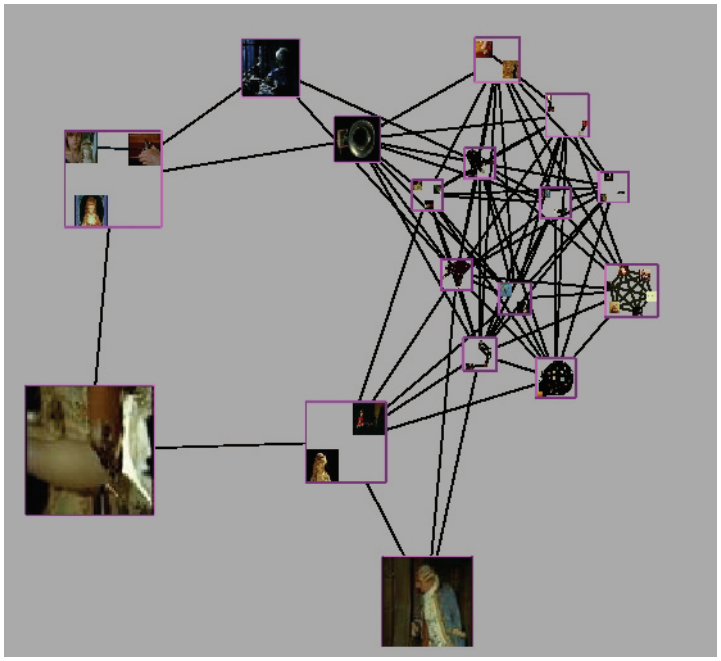


FIG. 2.9: Graphe quotient associé à la séquence "La joueuse de Tympanon" SFRS ©.

La figure 2.11 illustre la qualité des fragments produits par la méthode : la majorité des plans avec des contenus couleur similaires ont été groupés dans le même fragment. Ceci est lié au choix d'indexation des plans vidéo, basé sur la signature couleur spatio-temporelle ainsi qu'à la mesure de dissimilarité basée sur la distorsion de codage. La construction de la famille de DAG et le calcul du nombre de Strahler proposé dans cette section permettent de fragmenter le graphe en quasi-cliques.

L'algorithme de dessin des fragments utilisé est l'algorithme de dessin dédié aux DAG proposé par Sugiyama et al. [147]. Il permet de dessiner les sommets connectés de manière proche. De plus, la représentation des arêtes et l'affichage de l'image-clé associée au plan vidéo sur les sommets du graphe vidéo facilitent l'identification des fragments par l'utilisateur. Cette représentation est adaptée à la tâche qui consiste à évaluer la qualité des scènes vidéo construites.

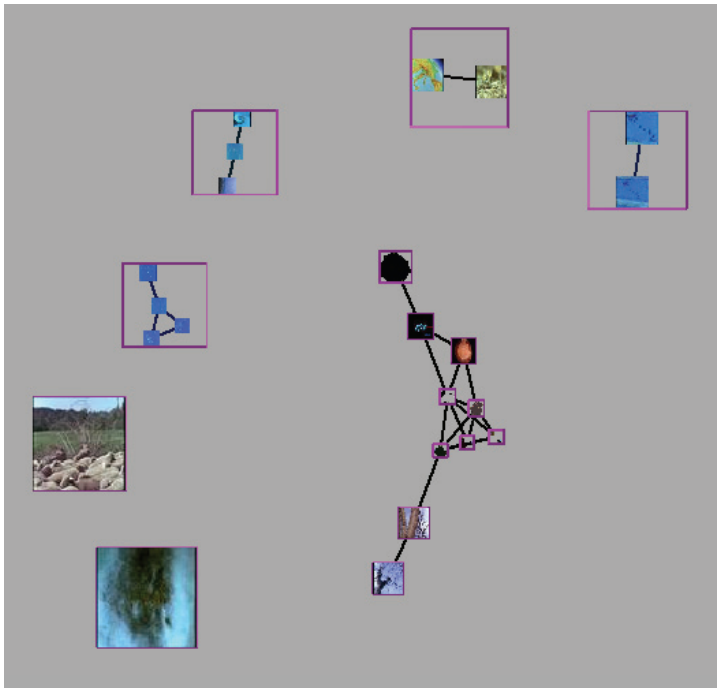


FIG. 2.10: Graphe quotient associé à la séquence “Chancre coloré du platane” SFRS ©.

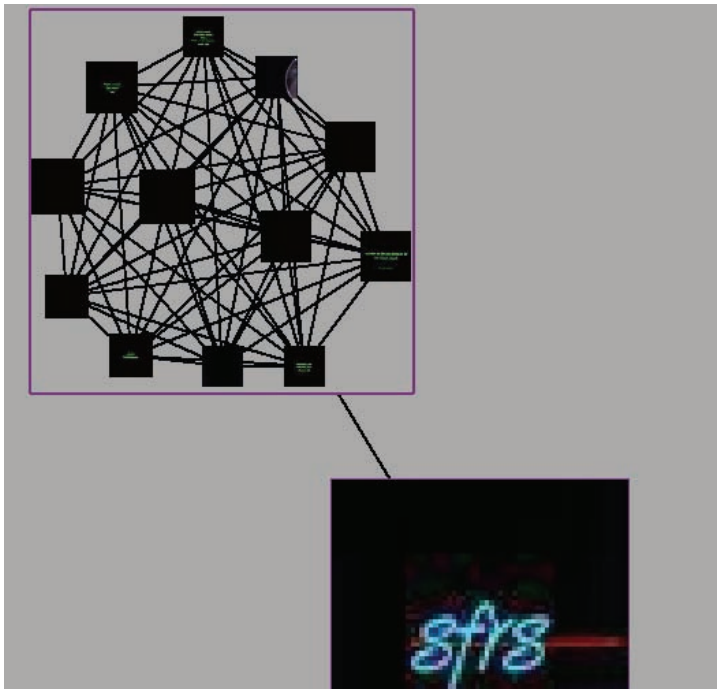


FIG. 2.11: Zoom géométrique sur un fragment de la séquence “Chancre coloré du platane” SFRS©.

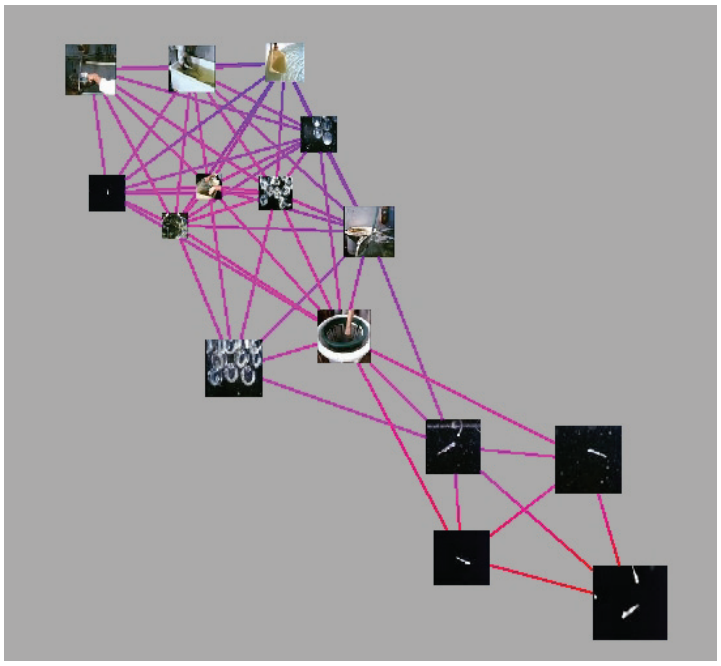


FIG. 2.12: Zoom géométrique sur un fragment de la séquence "Aquaculture en Méditerranée" SFRS©.

2.2.3 Complexité en temps

Proposition 2.1 (Complexité en temps)
Soit $G = (V, E)$ le graphe vidéo filtré associé au document vidéo composé de N plans indexés à l'aide du descripteur "X-Ray". L'ensemble des étapes de la méthode de fragmentation a une complexité en temps en $O(|N||E| \log |E|)$.

Pour évaluer la complexité en temps de la méthode, on doit considérer les différentes étapes. Initialement, le calcul des dissimilarités entre plans a une complexité en temps en $O(N^2)$. Le filtrage statistique du graphe est également en $O(N^2)$. Enfin, le calcul du nombre de Strahler de la racine de chacun des N DAG de la famille de DAG nécessite $O(|E| \log |E|)$, où E est l'ensemble d'arêtes du graphe vidéo après l'étape de filtrage. L'étape d'extraction des dictionnaires associés aux plans vidéo n'est pas prise en compte car c'est un prétraitement dont le résultat peut être réutilisé et stocké.

En pratique, les étapes de filtrage du graphe complet, le calcul des nombres de Strahler, et le dessin de graphe s'effectuent en moins de trois secondes sur un Pentium IV cadencé à 2.8 Ghz, et ce pour toutes les séquences présentées dans cette section. Ce délai est une valeur négligeable devant le temps utilisé pour la navigation dans le graphe quotient et l'examen des scènes produites.

2.3 Évaluation de la méthode

Nous proposons d'évaluer l'adéquation entre les fragments produits avec notre méthode, et une indexation manuelle des plans en scènes.

La caractéristique principale du partitionnement manuel de l'ensemble des plans vidéo en scènes, réside dans le fait que l'opérateur humain se base sur une *interprétation sé-*

mantique du contenu vidéo pour bâtir une scène. Cela peut introduire un biais lors de la comparaison des résultats produits de manière automatique ou interactive avec l'indexation de référence. La méthode proposée repose sur une indexation basée sur les couleurs des plans vidéo ; les scènes produites se caractérisent d'avantage par l'homogénéité de leur contenu couleur plutôt que par leur contenu sémantique. C'est un biais dont nous devrons tenir compte lors de la comparaison des fragmentations des plans en scènes d'un document vidéo.

2.3.1 Mesure de qualité des scènes

Le Tableau 2.2 présente les caractéristiques des indexations de référence associées aux quatre séquences utilisées (cf. Tableau 2.1). L'indexation des plans de contenu couleur similaire en hyper-scènes a été réalisée par un opérateur humain indépendamment du contexte de notre travail. La figure 2.13 montre un exemple d'hyper-scènes ainsi obtenues.

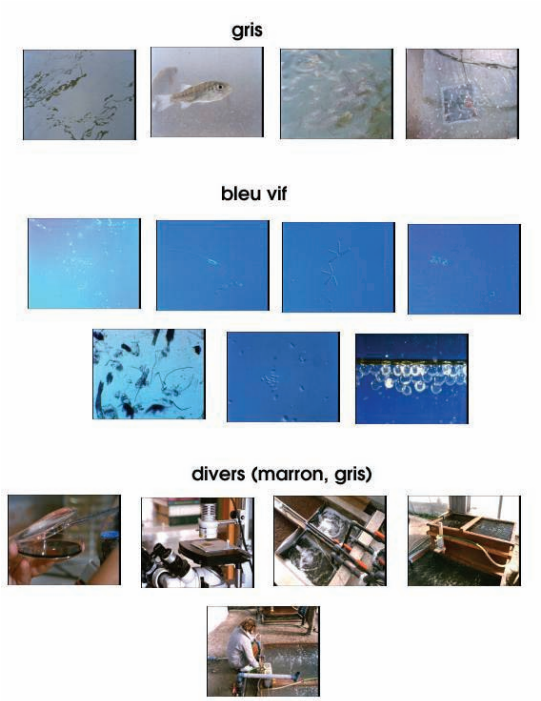


FIG. 2.13: Exemple d'hyper-scènes déterminées manuellement à partir du documentaire "Aquaculture en Méditerranée" SFRS©.

Le rappel et la précision (cf. Définition 7) sont calculés pour chaque scène détectée, et la moyenne sur l'ensemble des scènes est utilisée pour mesurer l'adéquation entre les scènes détectées avec la méthode présentée et les scènes de référence. Ces valeurs sont présentées dans le Tableau 2.3. Les chiffres de rappel sont assez bas. Cela peut s'expliquer par la nature de l'indexation de référence, basée sur une perception globale des couleurs dans les plans vidéo, et qui diffère de la notion de similarité exprimée par la signature spatio-temporelle des plans vidéo. Par ailleurs, lorsque le contenu couleur des plans constituant des hyper-

Nom de la séquence	Plans	Scènes
De l'arbre à l'ouvrage, SFRS©	321	12
La joueuse de Tympanon, SFRS©	200	7
Aquaculture en Méditerranée, SFRS©	87	19
Chancre coloré du platane, SFRS©	98	12

Tab. 2.2: Caractéristiques des indexations de référence.

scènes différentes est très proche, il semble que ni l'étape de filtrage, ni la fragmentation basée sur le nombre de Strahler ne parvienne à séparer ce type d'hyper-scènes.

Nom de la séquence	Précision Minimum	Précision Maximum	Précision Moyenne	Rappel Minimum	Rappel Maximum	Rappel Moyen
Chancre coloré du platane, SFRS©	20.0%	100.0%	73.2%	5.9%	92.9%	28.4%
La joueuse de Tympanon, SFRS©	0.0%	66.6 %	30.0%	0.0%	34.4%	13.2 %
De l'arbre à l'ouvrage, SFRS©	29.9%	100.0%	62.2%	1.2%	88.2%	16.1%
Aquaculture en Méditerranée, SFRS©	0.0%	100.0%	59.5%	0.0%	100.0%	43.0%

Tab. 2.3: Rappel et précision de l'indexation interactive des séquences vidéo traitées.

L'affichage du document sous forme de graphe quotient ainsi que la représentation des scènes sous forme de graphes permet d'identifier l'homogénéité d'un document ou d'une scène d'après la densité d'arêtes dans la zone correspondante du graphe. Ainsi, dans la figure 2.10, le graphe quotient de la séquence intitulée "Chancre coloré du platane" SFRS©n'est pas connexe car ce document a un contenu couleur très hétérogène composé d'un mélange de scènes d'intérieur, de vues microscopiques et de scènes d'extérieur. Par contre, comme le montre la figure 2.9, le graphe quotient associé aux scènes de "La joueuse de Tympanon" SFRS©présente une densité d'arêtes élevée, car les scènes de ce documentaire, principalement des scènes d'intérieur filmées dans un musée, sont perceptuellement homogènes. La même remarque s'applique pour la scène présentée à la figure 2.11 qui est une scène issue du générique de "Chancre coloré du platane" SFRS ©, où l'ensemble des plans qui composent cette scène sont des incrustations de texte sur un fond sombre. Le nombre d'arêtes qui relie les sommets de cette scène caractérise bien l'homogénéité de son contenu. A l'opposé, la figure 2.12 présente une scène dont la répartition des arêtes, moins homogène, reflète l'hétérogénéité des plans qui la composent.

2.4 Conclusion et perspectives

Dans ce chapitre, nous avons proposé une méthode de groupement des plans d'un document vidéo au contenu couleur similaire en hyper-scènes. Nous avons proposé de modéliser la similarité des plans vidéo du document par un graphe dont les arêtes sont valuées par

la mesure de dissimilarité entre plans. La signature spatio-temporelle des plans vidéo a été utilisée, car elle représente un bon compromis entre pouvoir discriminant et efficacité du calcul des dissimilarités [42].

Le filtrage des arêtes du graphe correspondant aux paires de sommets les plus dissimilaires est appliqué. Ce filtrage a pour effet :

- 1°, d'isoler des composantes connexes correspondant à des hyper-scènes,
- 2°, d'induire une structure de quasi-clique entre sommets appartenant à la même hyper-scène dans les composantes connexes correspondant à un mélange de plusieurs hyper-scènes.

Enfin, nous proposons d'utiliser le calcul du paramètre de Strahler sur une famille de DAG, pour isoler les sommets appartenant à la même quasi-clique et correspondant donc à des hyper-scènes du document vidéo analysé.

La méthode générale est paramétrée par :

- le nombre d'éléments dans les dictionnaires associés aux "bins" des projections "X-Ray" associés aux plans vidéo. Nous avons avons déterminé que seize "bins" par dictionnaire constituait un bon compromis entre la complexité en temps et l'efficacité de la signature spatio temporelle associée.
- La valeur Q_{max} correspondant au filtrage des arêtes du graphe. Nous proposons d'utiliser la valeur correspondant au maximum de l'histogramme des distances.
- La largeur du noyau de convolution utilisé pour déterminer les classes d'équivalence des sommets du DAG en fonction de leur valeur de Strahler. Cette valeur est fixée interactivement par l'utilisateur afin de choisir le meilleur réglage.

Les résultats obtenus sur quatre films documentaires ont été comparés avec une indexation manuelle de référence des plans vidéo en hyper-scènes homogènes au sens de la couleur. Les faibles valeurs de rappel indiquent, que dans certains cas la méthode proposée ne parvient pas à détecter certaines hyper-scènes. Cela correspond à des documents vidéo dont le contenu couleur varie peu de plan en plan.

L'utilisation d'autres paramètres calculés sur les sommets du graphe filtré pourraient permettre une meilleure identification des quasi-cliques. Ces quasi-cliques pouvant s'apparenter à la définition de communautés dans les réseaux sociaux, des métriques de graphe issues de ces travaux pourraient être utilisées à cet effet [7].

Chapitre 3
Indexation des scènes de dialogue

Dans ce chapitre, nous présentons une méthode d'indexation des scènes de dialogue visuelles issues d'une séquence vidéo naturelle [47]. Contrairement à la définition des scènes utilisée dans le chapitre 2, basée uniquement sur une similarité couleur de bas niveau entre les plans vidéo, les scènes de dialogue visuelles sont d'un niveau sémantique plus élevé puisqu'on cherche à identifier l'interaction entre deux personnages dans une succession alternée de plans vidéo de deux classes. Ce motif étant typique des scènes de dialogue, et dans la mesure où la technique de détection de ces motifs n'utilise pas l'information audio du document vidéo, on fait référence à ces scènes par le terme "scène de dialogue visuelle".

Ce type d'indexation de haut niveau sémantique est important pour au moins deux raisons :

- La détection automatique de scènes significatives dans les séquences vidéo facilite l'analyse et la comparaison de la structure des documents vidéo.
- Les frontières de scènes représentent des points d'accès privilégiés pour la navigation dans les documents vidéo.

De nombreuses méthodes ont été proposées pour la détection de scènes selon des définitions et des méthodes d'analyses différentes. On distingue les méthodes génériques des méthodes basées sur un modèle.

Dans les méthodes génériques, aucune hypothèse concernant le genre du document vidéo n'est utilisée. Dans [176], une séquence vidéo est segmentée en "Story Units". Des caractéristiques visuelles de bas niveau sont utilisées pour modéliser la similarité entre les plans vidéo. Une contrainte temporelle est ajoutée afin de limiter la construction de scènes aux seules suites de plans consécutifs. Les scènes produites sont des ensembles de plans visuellement similaires, et proches dans le temps. Dans cette approche, la granularité des scènes détectées dépend du seuil de similarité utilisé pour grouper les plans similaires et de la taille de la fenêtre temporelle qui définit la distance maximale entre 2 plans appartenant à la même scène.

Dans [148], des scènes de dialogue définies comme des suites d'au moins six plans vidéo suivant le motif $A - B - A - B - A - B$ sont détectées en utilisant une fonction de similarité, notée $\overline{\delta(n)}$, basée sur la norme L_1 entre les histogrammes de couleur associés à chaque plan vidéo. Cette fonction est calculée sur un intervalle de plans consécutifs et renvoie la similarité moyenne entre les plans situés toutes les n positions. Ainsi $\overline{\delta(0)}$ est la similarité moyenne entre le 1^{er} plan et lui-même, $\overline{\delta(1)}$ indique la similarité moyenne entre le plan courant et son successeur etc... En utilisant cette fonction, un test statistique est utilisé pour vérifier que $\overline{\delta(2)} > \overline{\delta(1)}$ et $\overline{\delta(2)} > \overline{\delta(3)}$ dans ce cas, l'intervalle de plans est identifié comme une scène de dialogue. Cette méthode ne fait pas appel à des règles basées

sur des informations sémantiques telles que la présence d'humains dans les plans. Toute scène périodique sera donc qualifiée de scène de dialogue.

Dans [128], les auteurs utilisent des algorithmes de détection de visage et de reconnaissance de visages pour détecter les scènes de dialogues selon le motif "shot/reverse-shot". Un système de classification basé sur des réseaux de neurones est utilisé pour détecter les visages humains [137]. Cette détection n'est appliquée qu'au zones de l'image correspondant à la couleur de la peau humaine. Ce processus est appliqué toutes les trois "frames" de chaque plan vidéo. Ensuite, les visages détectés dans chaque plan et partageant la même position et la même taille sont groupés dans la même classe de visage. Plusieurs classes de visages sont ensuite groupées selon la méthode de reconnaissance de visages "Eigenface" [157]. Une scène de dialogue est alors définie par une suite d'au moins trois plans vidéo contenant une classe de visage, et la même classe de visage apparaît tous les deux plans.

Les méthodes basées sur un modèle sont conçues pour extraire des scènes sémantiques de documents vidéo d'un genre spécifique : journaux télévisés, retransmissions sportives, longs métrages.

Dans [1], un modèle de Markov est utilisé pour la détection de scènes de dialogue. Ce modèle est paramétré grâce à un jeu de données d'apprentissage constitué de caractéristiques de niveau moyen issues du flux audio-visuel (classification audio en silence/musique/paroles, présence de visage dans les plans et caractéristiques couleur des plans.).

Dans [67], les auteurs utilisent la présence d'incrustations de logos dans les journaux télévisés pour différencier les séquences d'information des autres séquences (publicités). Ensuite, les régions d'intérêt sont localisées en utilisant la détection de la couleur de peau et la détection de logo. Ces informations sont utilisées par une procédure de mise en correspondance avec un modèle spatial afin de classifier les plans vidéo dans les catégories "présentateur" et "reportage". Enfin, des "News Units" sont construites pour structurer d'avantage la séquence vidéo.

Dans [150], les auteurs proposent des définitions strictes des différents types de scènes basées sur les règles de montage des films narratifs, et classifient les plans en "scènes avec mouvement", "scènes d'événements en série", et "scènes d'événements parallèles". L'algorithme identifie ces scènes en se basant sur la présence d'un fond commun entre différents plans, ou encore sur la présence d'objets en mouvement. Pour ce faire, un vecteur de caractéristique de bas-niveau est extrait de chaque plan. Il est composé des caractéristiques couleur de régions spécifiques de chaque "frame" : les quatre coins ainsi qu'une barre horizontale située en haut de la "frame".

Une approche complémentaire concernant la détection de scènes consiste à analyser conjointement les différents médias issus de la même séquence audiovisuelle, précisément le flux vidéo, le flux audio et les sous-titres. Dans le projet Informedia [75], les flux audio et vidéo sont analysés séparément pour définir deux ensembles de frontières de scènes. Des "paragraphes video" sont déterminés en utilisant une mesure de différence entre les histogrammes couleur associés aux "frames" consécutives. Concernant l'analyse du flux audio, les portions du signal de faible énergie sont considérées comme des silences, et forment les frontières des "paragraphes acoustiques". Dans une étape finale, les frontières audio et vidéo sont fusionnées pour définir les frontières de scènes globales.

La détection des scènes périodiques dans la succession naturelle des plans vidéo peut se faire directement sur la matrice de distance entre plans vidéo. Dans [125], une approche basée sur l'analyse spectrale de la matrice Laplacienne associée à un graphe est présentée. La matrice Laplacienne est construite à partir de la matrice de similarité entre les plans issus d'une séquence vidéo. Ce processus de fragmentation est utilisé pour grouper les plans

similaires en scènes.

La recherche de motifs dans la matrice de distance peut également être utilisée pour structurer les documents vidéo. Dans [38], les frontières de plans sont détectés en identifiant des blocs carrés le long de la diagonale principale de la matrice de similarité des "frames" vidéo.

La méthode présentée ici consiste à détecter les scènes de dialogue visuelles que l'on définit précisément *comme une succession de plans consécutifs suivant le motif $A-B-A...$, et dont chaque plan contient au moins un visage humain.* Seule l'information issue du flux vidéo est utilisée, et aucune hypothèse n'est faite concernant le genre du document vidéo traité. On utilise la matrice d'adjacence pondérée du graphe vidéo (cf. chapitre 2.1.3) correspondant au document vidéo analysé pour représenter les relations de similarité et l'enchaînement des plans vidéo.

La détection des motifs périodiques dans la succession des plans consiste à identifier des "motifs en damier" composés de 0 et de 1 sur la diagonale principale de la matrice de similarité après seuillage des valeurs. Les motifs sont isolés par des opérations de morphologie mathématique [141].

La seconde contribution de cette méthode concerne la coopération entre deux détecteurs de visages :

- un détecteur de visages basé sur les Machines à Vecteur de Support (SVM) et un détecteur basé sur l'utilisation d'un modèle de couleur de peau. Cette coopération vise à résoudre les problèmes suivants : d'abord, la variété des poses rencontrée dans les documents vidéo naturelles entraîne nécessairement une dégradation des performances d'un détecteur de visages basé sur l'apprentissage supervisé de visages de face.
- un détecteur de visages basé sur la couleur de peau permet la détection de nombreux "faux-positifs" qui doivent être filtrés : mains, bras et jambes ou objets ayant une couleur proche de celle de la peau. Enfin, les modèles de couleur de peau sont habituellement réglés à l'aide d'un ensemble de visages issus d'une collection d'apprentissage dont les caractéristiques colorimétriques et d'éclairage sont totalement différentes des images traitées. Notre approche permet d'entraîner un modèle de couleur de peau uniquement d'après les données vidéo issues du document analysé. Le modèle est donc particulièrement adapté aux teintes et à l'esthétique propres au document traité.

3.1 Formulation du problème

3.1.1 Scène de dialogue

Les scènes de dialogue visuelles que notre méthode vise à détecter suivent la définition suivante :

Définition 60 (Scène de dialogue visuelle)
Soit δ une mesure de dissimilarité entre plans vidéo. Dénotons par r, s, p, q, v, w des plans de la séquence de plans $P_i, P_{i+1}, ..., P_{i+k-1}$. La séquence de plans $P_i, P_{i+1}, ..., P_{i+k-1}$ est une scène de dialogue visuelle de longueur $k \geq 3$ si :
i) $\forall r, s \in \mathcal{A} = \{P_{i+d} | d \in [0, k-1], d \text{ est pair}\}$, $\forall v, w \in \mathcal{B} = \{P_{i+d} | d \in [0, k-1], d \text{ est impair}\}$, $\forall p \in \mathcal{A}, \forall q \in \mathcal{B}$, *les relations suivantes sont vraies :*

$$\delta(p, q) \gg \delta(r, s) \tag{3.1}$$

$$\delta(p, q) \gg \delta(v, w) \tag{3.2}$$

ii) $\forall r \in \mathcal{A} \bigcup \mathcal{B}$, *r contient au moins un visage.*

La première partie de la Définition 60 décrit le motif périodique dans l'enchaînement des plans dit en "champs/contre-champs". Dans certains cas, ce motif n'étant pas suffisant pour ne caractériser que les scènes de dialogue, la seconde partie de la définition permet de filtrer les scènes périodiques n'impliquant pas de personnages humains. Les scènes de génériques, ou les scènes montrant l'interaction entre un personnage et un objet d'intérêt en sont des exemples. La figure 3.1 montre un exemple de scène périodique qui n'est pas une scène de dialogue visuelle.

FIG. 3.1: Scène périodique qui n'est pas une scène de dialogue visuelle. Plans 134 à 137 du documentaire "Quel temps font-ils ?", SFRS©.

3.1.2 Résumé de la méthode

Comme le suggère la Définition 60, la méthode permettant de détecter les scènes de dialogue visuelle implique la coopération de plusieurs algorithmes. La figure 3.2 en résume les différentes étapes.

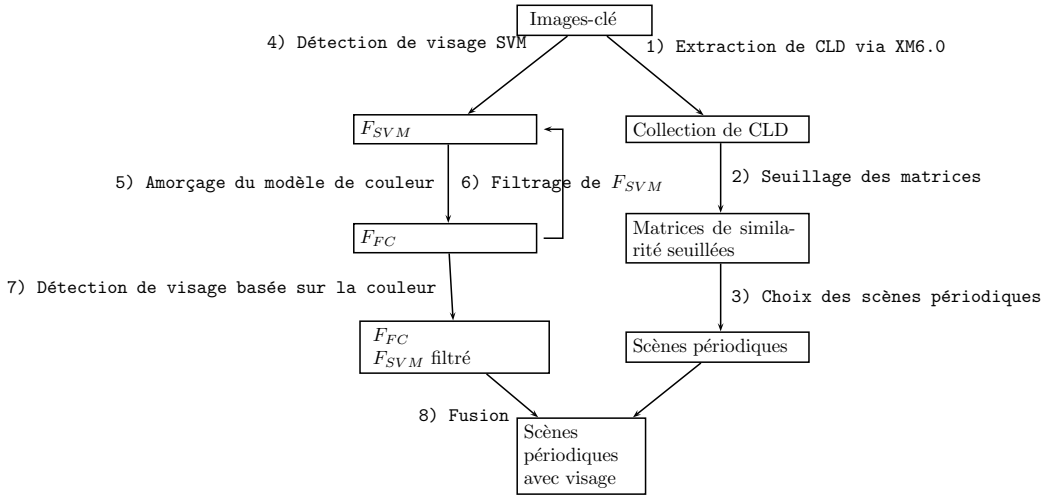


FIG. 3.2: Étapes de la méthode de détection des scènes de dialogue visuelles.

Après la détection des scènes périodiques, les deux ensembles constitués des positions des visages détectés avec les deux détecteurs sont utilisés pour filtrer les scènes périodiques qui ne correspondent pas à la deuxième partie de la définition 60. La méthode de détection par SVM a été mise au point dans la thèse de L. Carminati [29]. Soit F_{SVM} l'ensemble des visages détectés par l'algorithme basé sur les Machines à Vecteur de Support et soit F_{FC} l'ensemble de visages détectés par la méthode basée sur le modèle de couleur de peau.

L'intersection entre ces deux ensembles est vide, $F_{SVM} \bigcap F_{FC} = \emptyset$, car on n'utilise les visages de F_{FC} que pour compléter les résultats du détecteur à SVM.

La boucle impliquant les étapes 5 et 6 dans la figure 3.2 représente la coopération entre les 2 détecteurs de visages. C'est dans cette boucle qu'est amorcé le modèle de couleur de peau à partir des informations de couleur issues des visages de F_{SVM}. A l'étape 6, le modèle de couleur est utilisé pour filtrer les visages de F_{SVM} dont la majorité des pixels n'appartient pas au modèle de couleur de peau. Les résultats expérimentaux montrent que cette approche permet de filtrer la plupart des faux-positifs initialement présents dans F_{SVM}.

A l'étape 7, le modèle de couleur de peau final construit à partir du dernier ensemble de visages F_{SVM} filtré est utilisé pour construire l'ensemble de nouveaux visages F_{FC}.

3.2 Détection des scènes périodiques

3.2.1 Construction de la matrice de similarité

On suppose que la segmentation de la séquence vidéo en plans a été effectuée. On peut utiliser à cet effet l'une des méthodes récentes développées dans le cadre de la campagne d'évaluation TREC video [123], dont les performances atteignent aujourd'hui une précision et un rappel proches de 100%. Si le problème de la segmentation du document en plans de montage n'est pas abordé ici, cette étape est néanmoins cruciale pour le bon déroulement des étapes suivantes. En effet, une sous-détection ou une sur-détection d'une frontière de plan détruit immanquablement la périodicité des plans dans la scène. En pratique, une indexation manuelle des frontières de plans est utilisée.

Une image-clé est ensuite extraite de chaque plan et va être utilisée pour caractériser le plan entier. On n'aborde pas le problème complexe de la sélection de l'image-clé. L'image située au milieu de chaque plan est choisie comme image-clé. Ceci conduit à l'ensemble $K = \{k_i\}_{1 \leq i \leq N}$ composé de N images-clé.

Le descripteur "Color Layout Descriptor" (CLD), issu du standard MPEG7 [34], est extrait de chacune des images-clé. Soit $F = \{CLD_i\}_{1 \leq i \leq N}$ l'ensemble des descripteurs extraits. Ce descripteur est utilisé, car il permet de caractériser les couleurs présentes dans l'image ainsi qu'une partie de la disposition des couleurs dans l'image. Ce dernier point est important dans le contexte de la détection des scènes périodiques puisqu'on doit pouvoir distinguer les plans d'une scène de dialogue présentant deux personnages filmés sur un fond similaire et dans les mêmes conditions d'éclairage. Dans ce cas, un descripteur basé uniquement sur les couleurs, tel que le descripteur MPEG7 "Dominant Color Descriptor", ne permettrait pas de différencier les deux types de plans et rendrait impossible la détection de la scène périodique.

A partir de l'ensemble F, on construit la matrice de dissimilarité D, de taille $N \times N$ en utilisant la mesure de dissimilarité associée au CLD, notés δ_{CLD} :

$$D = (\delta_{CLD}(CLD_i, CLD_j))_{1 \leq i \leq N, 1 \leq j \leq N} \tag{3.3}$$

La matrice D est ensuite normalisée et transformée en la matrice de similarité M :

$$M = (1 - \frac{D_{ij}}{\max(D)})_{1 \leq i \leq N, 1 \leq j \leq N} \tag{3.4}$$

D'après la définition de δ_{CLD}, M est symétrique et contient la valeur 1 sur sa diagonale principale.

3.2.2 Détection de motifs périodiques à un niveau de seuillage

L'ensemble de plans consécutifs qui satisfait la première partie de la Définition 60 peut être identifié visuellement par les motifs "en damier" qui apparaissent le long de la diagonale principale de la matrice M.

Un motif "en damier" correspond au motif formé par une suite d'au moins trois plans consécutifs respectant la première partie de la Définition 60 dans une visualisation de la matrice de similarité des plans vidéo.

La figure 3.3 présente une telle visualisation avec un motif "en damier" impliquant trois plans. Une case noire représente une similarité minimale, et une case blanche correspond à une similarité maximale. La diagonale du motif est nécessairement blanche, car ses cases correspondent à la similarité d'un plan avec lui-même, laquelle est maximale. Les cases correspondant à la similarité entre P_i et P_{i+2} ne sont pas blanches mais correspondent néanmoins à une forte similarité. En revanche, P_{i+1} n'est similaire ni avec P_i ni avec P_{i+2}, ce qui se caractérise par quatre cases foncées.

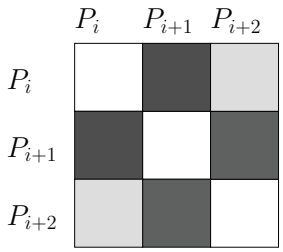


Fig. 3.3: Exemple de motif "en damier" impliquant trois plans.

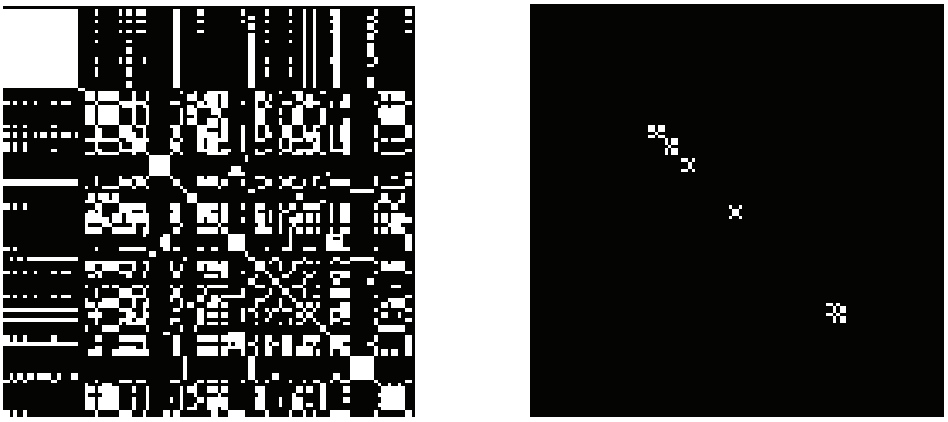
Afin d'isoler ces motifs, on va transformer la matrice M en une image binaire par seuillage et utiliser les outils morphologiques pour en extraire les motifs recherchés. Soit $t \in [0;1]$ un seuil, la matrice M est transformée en image binaire I_t par un seuillage de niveau t :

$$I_t(i,j) = \begin{cases} 1 & \text{if } M_{i,j} > t \\ 0 & \text{if } M_{i,j} \leq t \end{cases} \tag{3.5}$$

Après un tel seuillage, des blocs constitués de 1 autour de la diagonale principale correspondent à des suites de plans similaires. Ces blocs ne correspondent pas à notre définition de scène et doivent donc être supprimés. On remplace ces blocs de 1 par des 0, ce qui conduit à une nouvelle image I'_t. Les motifs périodiques recherchés s'apparentent à des motifs en forme de "X" dans I'_t. On utilise une technique de filtrage morphologique [141] sur I'_t pour localiser ces éléments. Une ouverture morphologique est appliquée à I'_t en utilisant un élément structurant en forme de "X" de taille 3×3, noté X. Cette opération conduit à :

$$I''_t = I'_t \circ X. \tag{3.6}$$

La figure 3.4 illustre le processus de filtrage morphologique. Les scènes périodiques sont identifiées par des blocs carrés composés d'un motif en damier le long de la diagonale principale de I''_t.



a) $I_{0.7}$ b) $I''_{0.7}$

FIG. 3.4: Exemple de filtrage morphologique d'une image binaire avec un élément structurant en forme de "X"de taille 3×3. a) Image binaire correspondant à un seuillage de la matrice de distance à $t = 0.7$, b) Résultat de la suppression des blocs carrés et de l'ouverture morphologique.

On note l'ensemble des scènes périodiques détectés dans I''_t

$$\mathcal{S}_t = \{S_{t,j,k}\} \tag{3.7}$$

$$S_{t,j,k} = (P_i)_{j \leq i \leq j+k-1} \tag{3.8}$$

3.2.3 Persistance des motifs périodiques

En fonction de la valeur de seuil t choisie, la détection des scènes périodiques peut varier de manière significative. Supposons que t ait une valeur trop élevée, certaines valeurs égales à 1 contribuant à un motif périodique seraient alors transformées en 0, et le motif disparaîtrait. En diminuant progressivement la valeur de t, des valeurs "manquantes" égales à 1, apparaîtront en complétant les motifs périodiques.

La situation réciproque peut être observée si le seuil initial est trop bas. Dans ce cas, certains éléments de valeur 0 n'apparaîtront pas, et les motifs périodiques seront transformés en carrés de 1. En augmentant progressivement la valeur de t, les motifs périodiques apparaîtront progressivement.

Ces remarques indiquent qu'il n'est pas souhaitable d'utiliser une valeur de seuillage unique pour la détection des motifs périodiques. En effet, en fonction du contenu des images des deux classes intervenant dans un motif périodique, différentes scènes périodiques ne pourraient pas être détectées simultanément avec la même valeur de seuil. Par exemple, une scène de dialogue dont la similarité entre la classe des plans de type A et celle des plans de type B serait élevée, pourrait ne pas être détectée en utilisant la même valeur de seuil t que pour une scène pour laquelle le niveau de similarité à l'intérieur des classes A et B serait inférieure à t.

D'après la première partie de la Définition 60, on souhaite conserver les motifs périodiques où le contraste entre les plans de type A et les plans de type B est élevé tout en autorisant une tolérance à l'intérieur de chaque classe A et B. Ainsi, nous proposons une méthode de détection basée sur la persistance des motifs périodiques détectés avec des seuils de plus en plus faibles.

Fixons une valeur de seuil initiale élevée, notée t_1, et diminuons progressivement cette valeur jusqu'à un seuil t_q. A chaque valeur de seuil correspond une matrice de similarité seuillée. Afin de détecter des scènes périodiques suffisamment contrastées en termes de similarité, on utilise un paramètre qui désigne le nombre minimal de niveaux de seuillage consécutifs dans lesquels un motif doit exister. Supposons qu'un motif périodique soit détecté au niveau de seuillage $t_\alpha < t_1$, et disparaisse au niveau t_β tel que $t_q \leq t_\beta \leq t_\alpha$. La longueur de l'intervalle $t_\alpha - t_\beta$ exprime la persistance du motif périodique. On utilisera cette notion de persistance comme un paramètre de la méthode que l'on désignera par m. Soit $t_1 ... t_q$ une séquence de seuils telle que $t_i > t_{i+1}$. L'ensemble de scènes périodiques, noté $\mathcal{S}$ est :

$$\begin{array}{l} \mathcal{S} = \{Last(S_{t_\alpha,j_\alpha,k_\alpha}, .., S_{t_\beta,j_\beta,k_\beta})\} \\ \text{tel que,} \\ i)\ S_{t_i,j_i,k_i} \subseteq S_{t_{i+1},j_{i+1},k_{i+1}} \\ ii) \frac{t_\alpha - t_\beta}{th_1 - th_q} * 100\% \geq m\% \end{array} \tag{3.9}$$

Ici, $Last(e_1, e_2, ..., e_k)$ représente l'opérateur de sélection du dernier élément de la suite $e_1, e_2, ..., e_k$. Une scène périodique correspond donc au dernier élément d'une série de motifs périodiques de taille croissante, détectés à des niveaux de seuillage t décroissants et qui existent sur $m\%$ de l'intervalle de seuillage.

Une fois détecté à un niveau de seuillage t_α, la taille d'un motif périodique croit pour des niveaux de seuillage plus bas. En diminuant le niveau de seuillage, on accepte une similarité plus faible entre les 2 classes de plans A et B. Ainsi, afin de maximiser la taille des scènes détectées, le dernier élément de la série est conservé. La variabilité permise entre les plans à l'intérieur des ensembles de plans A et B est contrôlée par la valeur du niveau de seuillage le plus bas t_q. Les valeurs de t_1, t_q et m ont été déterminées expérimentalement d'après les résultats obtenus pour différents types de contenus vidéo : $m = 20\%$, $t_1 = \max(M)$ et t_q tel que 80% des valeurs de M soient supérieures à t_q.

3.3 Détection de visage

La détection de visages dans des séquences vidéo naturelles, requiert une méthode capable de fonctionner sur des scènes complexes. Les méthodes de détection de visages constituent un domaine de recherche très actif [174]. La méthode proposée ici, fait appel à deux types de détecteurs de visages. Le premier est basé sur les machines à vecteurs de support (SVM) et le second sur un modèle de couleur de peau. L'originalité de la méthode réside dans la coopération des deux détecteurs dans une boucle d'amorçage du modèle de couleur de peau et de filtrage des visages détectés par SVM. Le détecteur de visage SVM détecte un sous ensemble des visages humains présents dans les images-clé issues du document vidéo. Un modèle de couleur de peau est ensuite amorcé à partir du résultat précédent, considéré comme une donnée d'apprentissage.

3.3.0.1 Détection par SVM

Les détecteurs de visages basés sur les machines à vecteur de support ont été utilisés avec succès dans de nombreux domaines et plus particulièrement dans celui de la vidéo surveillance [29]. Les SVM constituent un puissant outil pour résoudre des problèmes de classification supervisée en deux classes. Ici, on expose brièvement la méthode mise au point dans [29].

Supposons que l'on souhaite estimer une fonction $f : R^n \longrightarrow \{\pm 1\}$ en utilisant un ensemble d'apprentissage $(x_1, y_1), ..., (x_l, y_l) \in R^n \times \{\pm 1\}$ constitué de couples dont le premier élément est le signal d'apprentissage et le second élément est l'étiquette associée à ce signal. On souhaite que f classifie correctement des exemples inconnus (x, y), c'est à dire que $f(x) = y$ pour des exemples produits à partir de la même distribution de probabilité que les données d'apprentissage, notée $\mathbb{P}(x, y)$. Cela signifie que les données doivent être séparées en deux classes. Même si la fonction produit de bons résultats sur l'ensemble d'apprentissage, par exemple $f(x_i) = y_i,\ \forall i = 1, .., l$, il n'est pas garanti que les performances se généralisent pour des exemples inconnus.

Dans la théorie de la classification, une fonction de classification f est choisie de telle sorte que l'erreur de classification, ou risque empirique, noté R_{emp}, commis sur l'ensemble d'apprentissage soit minimal :

$$R_{emp}(f) = \frac{1}{2l} \sum_{i=1}^{l} |y_i - f(x_i)| \tag{3.10}$$

Cependant, cela ne garantit pas que l'erreur de classification sur un ensemble inconnu, ou risque, soit également minimal (étape de généralisation).

La théorie des SVM [161] montre qu'un petit nombre de points d'intérêt peut être utilisé dans l'étape de généralisation. Ces points, situés près de l'hypersurface qui délimite les deux classes dans l'espace de décision, sont appelés vecteurs de support.

Dans le contexte de la détection de visages dans la vidéo, le problème est exprimé de la manière suivante. On considère des fenêtres de taille fixe construites à partir des données issue de l'image. Ces fenêtres peuvent correspondre ou non à l'emplacement d'un visage dans l'image. L'étape d'apprentissage d'une machine à vecteurs de support, consiste à construire l'hypersurface de classification à partir d'exemples étiquetés. L'étape de généralisation consiste à classifier des fenêtres issues des images-clé de la vidéo en deux classes : "visage" et "non-visage".

L'apprentissage est réalisé en sélectionnant et en étiquetant des fenêtres de $N \times N$ pixels contenant des visages et des portions de l'arrière plan (les représentants de la classe "non-visage) issues de "frames" vidéo en pleine résolution. L'étape de généralisation consiste à parcourir une image inconnue avec une fenêtre de taille $N \times N$, appelée rétine [29], et à classifier chaque fenêtre dans la catégorie "visage" ou "non-visage". Afin d'accroître la robustesse de cette méthode vis à vis de la taille des visages qu'il est possible de détecter, un mécanisme de multi-résolution est utilisé [29]. Le parcours de l'image à traiter par la rétine est effectué du niveau le plus haut (basse résolution) au niveau le plus bas (pleine résolution) d'une pyramide multi-résolution.

3.3.0.2 Amorçage du modèle de couleur de peau

Dans cette section, on présente le processus qui met en jeu la coopération entre le détecteur SVM et un modèle de couleur de peau. Cette étape permet de paramétrer le modèle de couleur de peau à partir des données issues du document vidéo à traiter, et de filtrer les "faux-positifs" issus de la phase de détection par SVM.

Modèle de couleur de peau Afin de modéliser la couleur de peau, il faut choisir un espace de couleur approprié, et identifier des fragments associés à la couleur de peau dans cet espace. On propose d'utiliser l'espace de couleur $YCbCr$ pour représenter le modèle de couleur de peau, car c'est un espace perceptiblement uniforme en comparaison avec

l'espace RGB. De plus, l'information de luminance et celle de chrominance est décorrelée. La première propriété permet d'utiliser une mesure de distance identique en tout point de cet espace couleur, et la seconde propriété permet de représenter les nuages de points associés aux couleurs dans un repère dont les axes sont alignés avec les directions principales des nuages de point.

Soit un échantillon $X = (x_1, .., x_n)^T$ de couleurs de peau composé de n valeurs de couleur dans l'espace $YCbCr$. On modélise la distribution des couleurs de l'échantillon par un modèle $\Phi = (w_1, .., w_K, \theta_1, .., \theta_K)$, composé d'un mélange de K distributions Gaussiennes tridimensionnelles de paramètre $\theta_k = (\mu_k, \sigma_k)$, où μ_k désigne le vecteur couleur moyenne, σ_k la matrice de covariance, et les w_k les proportions des différentes populations représentés. La loi que suit le mélange peut s'écrire :

$$g(x, \Phi) = \sum_{i=1}^{K} w_i f(x, \theta_i)$$

avec $f(x, \theta_i)$ la loi normale tridimensionnelle paramétrée par θ_i et μ_i.

On utilise un mélange de distributions Gaussiennes afin de pouvoir prendre en compte plusieurs groupes de teintes de peau. C'est important dans le contexte des vidéo naturelles pour lesquelles les conditions d'illumination et l'apparence colorée de la peau peuvent varier le long du document et même au sein d'une "frame".

Après la phase de détection de visage par SVM, la phase d'amorçage du modèle de couleur démarre. Ici, nous avons utilisé la méthode d'apprentissage des mélanges de Gaussiennes, proposée dans [29] pour la détection du mouvement dans une séquence d'images. Cet amorçage est basé sur l'estimation des paramètres Φ de la mixture Gaussienne. L'échantillon de couleur de peau est constitué à partir des zones de l'image issues de F_{SVM}. On suppose que la majeure partie de ces visages est composée de pixels de couleur de peau ; on considère donc tous les pixels situés au centre de ces zones comme faisant partie de l'échantillon.

Une estimation initiale des paramètres Φ est donnée par l'algorithme de fragmentation non-supervisé ISODATA [11] qui détermine également automatiquement le nombre de groupes K présents dans l'échantillon. L'algorithme EM [44] est ensuite utilisé pour maximiser la vraisemblance des paramètres des modèles Gaussiens étant donné l'échantillon.

Test de décision statistique Afin de déterminer si une couleur x appartient au modèle de couleur de peau, on doit déterminer quelle est la Gaussienne la plus vraisemblable. Pour cela, nous utilisons la méthode présentée dans [29].

La vérification est réalisée de la façon suivante : soit x un échantillon de couleur, nous cherchons à maximiser la vraisemblance de la composante η_i dans le mélange $p(x) = \sum_{i=1}^{K} w_t.\eta(x|\theta_i)$ par rapport à x. Dans ce cas, nous allons considérer la vraisemblance de la composante η_i conditionnellement à l'ensemble du mélange.

Soit une partition de l'espace des hypothèses H définit par $B = \{H_1, ..., H_i, ..., H_K\}$. Associons chaque loi Gaussienne η_i avec H_i. En considérant le théorème de Bayes, nous avons $P(H_i/B) = P(H_i.B)/P(B)$. En supposant que H_i, $i = 1, ..., K$ forme une partition complète de B, c'est à dire que $B = H_1 \cup ... \cup H_i \cup ... \cup H_k$, avec $P(B) = 1$ et $P(H_i.H_j) = 0$. Ainsi, $P(H_i.B) = P(H_i.(H_1 + H_2 + ... + H_i + ... + H_K)) = P(H_i)$. Cela se déduit de l'indépendance des hypothèses $\forall i \neq j$, $P(H_i.H_j) = 0$ et du fait que $P(H_i.H_i) = P(H_i)$. Comme $P(B) = 1$, on a donc $P(H_i/B) = P(H_i)$.

La densité de probabilité $p(x \in H_i/B)$ est telle que

$$p(x \in H_i/B) = \frac{w_i \eta_i}{\sum_{i=1}^{K} w_i \eta_i} \tag{3.11}$$

alors la vraisemblance conditionnelle de la i^{me} Gaussienne dans le mélange pour un échantillon x est

$$l(x) = \frac{w_i \eta_i}{\sum_{i=1}^{K} w_i \eta_i} \tag{3.12}$$

Suivant le processus de décision usuel, on cherche à maximiser la *log-vraisemblance* $L_i = \log(l(x))$. Nous obtenons ainsi

$$L_i = \log(l(x)) = \log \frac{w_i \eta_i}{\sum_{i=1}^{K} w_i \eta_i} = \log(w_i \eta_i) - \log(\sum_{i=1}^{K} w_i \eta_i). \tag{3.13}$$

or comme $\log(\sum_{i=1}^{K} w_i \eta_i)$ est le même pour tous les L_i, cela revient à maximiser $\log(w_i \eta_i)$ exprimé de la façon suivante :

$$\log(w_i \eta_i) = \log w_i + \log(f(x, \theta_i)) \tag{3.14}$$

ce qui équivaut à rechercher η_i^* et $\theta_i^* = (w_i^*, \mu_i^*, \sigma_i^{2*})$ tels que

$$\theta_i^* = \underset{\theta_i = (w_i, \mu_i, \sigma_i^2)}{\operatorname{argmax}} (\log w_i + \log(f(x, \theta_i))) \tag{3.15}$$

Ainsi, la "meilleure" loi Gaussienne pour un échantillon donné x peut être sélectionnée selon cette méthode. On effectue ensuite un test d'appartenance de x à l'intervalle de confiance à 1.5σ de la Gaussienne sélectionnée afin de déterminer si x appartient au modèle de couleur de peau.

Amorçage et filtrage La méthode d'amorçage du modèle de couleur a pour but principal la détection de nouveaux visages dont la pose n'est pas faciale. De plus, on propose d'utiliser le modèle de couleur de peau afin de filtrer l'ensemble F_{SVM} pour en supprimer les visages dont la majorité des pixels n'appartient pas au modèle de couleur.

La bonne précision de l'algorithme de détection par SVM implique qu'il existe plus de vrais visages que de faux positifs dans la version initiale non filtrée de F_{SVM}. De plus, on constate que la majeure partie des faux positifs est constituée de zones de petite taille, correspondant à des zones texturées de l'image contenant des motifs ressemblant à des visages. Ainsi, les faux positifs ont une taille inférieure à la plupart des vrais visages et leur nombre est inférieur à celui des vrais visages. On suppose que les valeurs associées aux faux-positifs vont contribuer à augmenter la variance des distributions Gaussiennes des différents groupes du modèle de couleur. Les valeurs de couleur issues des faux-positifs doivent donc se trouver hors de l'intervalle de confiance des modèles Gaussiens.

Une fois le modèle de couleur amorcé à l'aide de l'échantillon d'apprentissage issu de F_{SVM}, les visages de F_{SVM} dont la majorité des pixels n'appartient pas au modèle de couleur sont filtrés en tant que faux-positifs. Une fois ce filtrage effectué, le processus d'amorçage du modèle de couleur reprend sur l'ensemble F_{SVM} filtré. Ce processus est répété jusqu'à ce qu'aucun visage ne soit plus filtré de F_{SVM}.

Cette boucle d'amorçage-filtrage augmente la précision des résultats produits par l'algorithme basé sur les SVM et permet également d'adapter le modèle de couleur sur les

données du modèle de document uniquement. Le modèle sera donc mieux adapté à la nature et aux conditions d'éclairage du document vidéo que dans le cas où un modèle de couleur de peau aurait été entraîné sur un échantillon d'apprentissage issu de documents différents.

Soit F_{GT} l'ensemble de visages correspondant à la localisation parfaite de tous les visages. La précision, notée P, et le rappel, noté R sont donnés par :

$$P = \frac{|F_{SVM} \cap F_{GT}|}{|F_{SVM}|}$$

$$R = \frac{|F_{SVM} \cap F_{GT}|}{|F_{GT}|}$$

La figure 3.5 montre l'évolution de la précision et du rappel associés à l'ensemble de visages de F_{SVM} au cours du processus d'amorçage-filtrage sur le documentaire "Quel temps font-ils ?", SFRS©. L'ensemble F_{GT} a été constitué manuellement et contient les caractéristiques des rectangles englobant un visage de face ou de profil (cordonnées du coin supérieur gauche, largeur et hauteur). Un total de 456 visages a ainsi été déterminé.

Chaque image-clé a été traitée par un détecteur de visages basé sur les SVM, entraîné sur une base de 6977 visages de face et 23478 exemples de la classe "non-visage". L'ensemble F_{SVM} initial contenait 191 visages, avec une précision de 49% (94 vrais visages ; 97 faux-positifs) et un rappel de 21%. La valeur de rappel n'est pas représentative de la qualité du détecteur de visages car F_{GT} inclue des visages non frontaux.

Après 7 itérations de la boucle d'amorçage-filtrage, l'ensemble finale F_{SVM} contient 101 visages, avec une précision de 87% (88 vrais visages ; 13 faux-positifs) et un rappel de 19%. Parmis les 13 faux-positifs conservés dans F_{SVM}, 7 correspondent à des zones de l'image dont la couleur est proche de celle de la peau qui ne peuvent donc pas être rejetées selon le critère de la couleur. La figure 3.5 indique que la méthode améliore nettement la précision, sans trop sacrifier au rappel, c'est à dire sans supprimer trop de vrais visages de F_{SVM}. En effet, seuls $6,38\%$ de vrais visages ont été rejetés par le processus de filtrage.

3.3.0.3 Détection de visages basée sur le modèle de couleur de peau

Le détecteur basé sur les SVM ne fournissant que des *visages de face*, le rôle du modèle de couleur de peau est de permettre la détection d'un ensemble de nouveaux visages, *non-frontaux*, noté F_{FC}.

Dans les images-clé qui ne contiennent pas déjà un visage déterminé par SVM, les pixels qui appartiennent au modèle de couleur de peau sont étiquetés de manière à obtenir une carte de segmentation initiale.

Afin de détecter les zones composées d'étiquettes connexes de la carte de segmentation, le traitement suivant est effectué :

1. Filtrage médian afin d'éliminer les étiquettes isolées.
2. Dilatation morphologique avec un élément structurant carré de taille 5×5 afin de reconnecter les régions sur-segmentées.
3. Pour chaque région 4-connexe, sa boite englobante est calculée et les boites s'intersectant sont fusionnées.

Pour une image-clé donnée, on forme ainsi un ensemble de régions candidates F_{FC}. Néanmoins, ces régions peuvent ne pas correspondre à de vrais visages mais à d'autres parties du corps ou à des zones colorées comme de la peau. Afin d'éliminer ces régions, on applique plusieurs règles de filtrage différentes sur F_{FC} :

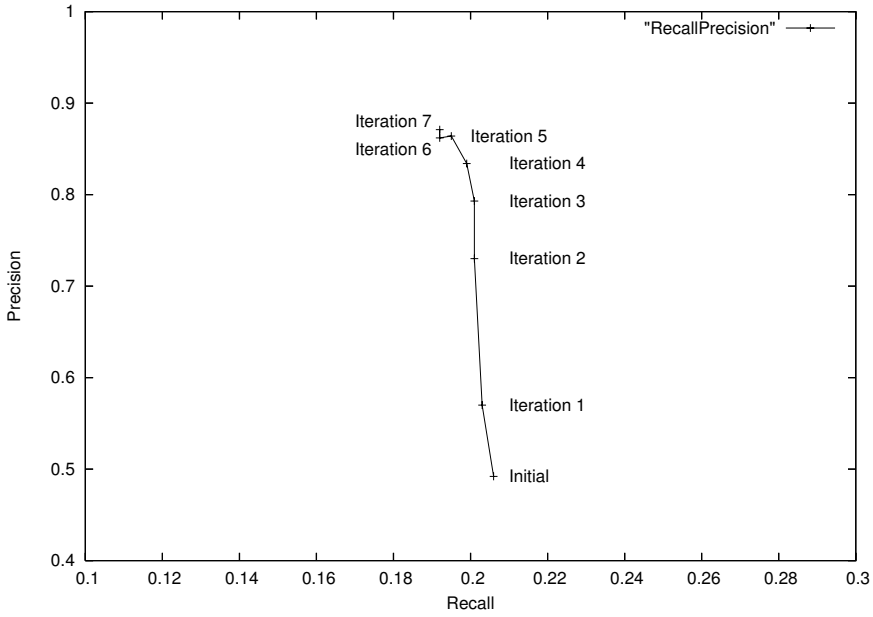


FIG. 3.5: Précision et rappel associés à l'ensemble F_{SVM} au cours des 7 itérations de la boucle d'amorçage-filtrage.

1. *Règle de localisation* : les visages issus du même document vidéo se situent dans la même région de l'image. La distribution de probabilité associé à la position du centre d'un visage dans l'image est modélisée par une Gaussienne bidimensionnelle. Les paramètres de la Gaussienne μ et σ sont déterminés d'après les positions des centres des visages de l'ensemble F_{SVM} filtré. Les visages de F_{FC} dont les coordonnées du centre n'appartiennent pas à la distribution sont rejetés.

2. *Règle de la taille minimum* : les boites englobantes issues de F_{FC} dont la hauteur est plus petite que la hauteur de visage minimale dans F_{SVM} sont rejetées.

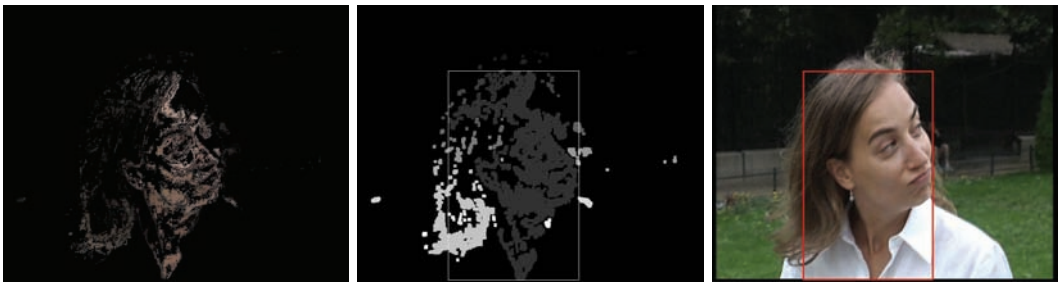
3. *Règle de la taille maximum* : idem. Cette règle vise à rejeter les grandes régions constituées de pixels appartenant à l'arrière-plan de l'image et qui auraient été étiquetés comme appartenant au modèle de couleur de peau.

Ces règles de filtrage sont totalement paramétrées par les caractéristiques des boites issues de F_{SVM}, telles que leur taille et position dans l'image. Encore une fois, cette étape est adaptée aux caractéristiques du document analysé.

Après filtrage, les régions de F_{FC} sont conservées comme un ensemble de visages complémentaire de F_{SVM}.

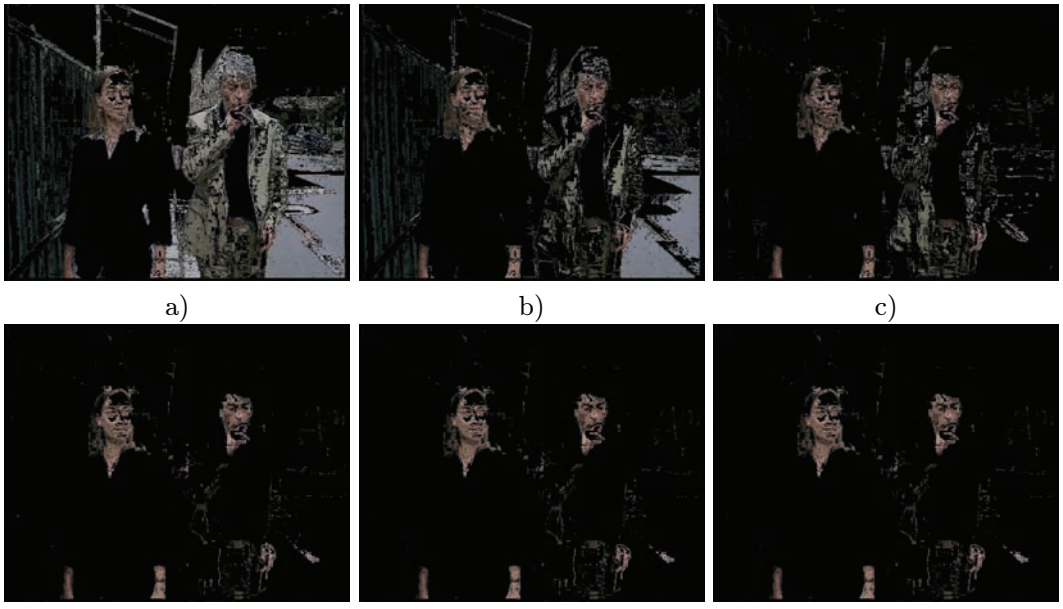
La figure 3.6 illustre les différentes étapes de la détection de visage par utilisation du modèle de couleur de peau.

La figure 3.7 illustre le raffinement du modèle de couleur de peau au cours des itérations de la boucle d'amorçage-filtrage. On constate que de moins en moins de pixels associés à l'arrière-plan sont considérés comme appartenant au modèle de couleur de peau.



a) b) c)

FIG. 3.6: Étapes de détection de visage basée sur le modèle de couleur de peau. a) Étiquetage des pixels appartenant au modèle de couleur de peau. b) Détection et filtrage des boites englobantes. c) Résultat final.



a) b) c)

d) e) f)

FIG. 3.7: Étiquetage des pixels appartenant au modèle de couleur de peau au cours de différentes itérations de la boucle d'amorçage-filtrage. a) Modèle initial. b) Itération 1. c) Itération 2. d) Itération 3. e) Itération 4. f) Itération 5.

3.4 Évaluation de la méthode

A l'issue des phases de détection de visages et de détection des scènes périodiques, seules les scènes périodiques dont chaque plan contient au moins un visage sont conservées. L'ensemble final de scènes périodiques, noté $\mathcal{S}_{final}$ est défini par :

$$\begin{aligned} \mathcal{S}_{final} &= \{S_{t,j',k'} \subseteq S_{t,j,k} \in \mathcal{S} | && (3.16)\\ & k' \geq 3, && (3.17)\\ & \forall P_i \in S_{t,j',k'}, && (3.18)\\ & P_i \textit{ contient un visage de } F_{SVM} \cup F_{FC}\} && (3.19) \end{aligned}$$

On présente les résultats expérimentaux obtenus sur un documentaire de 406 plans au format CCIR601 (720x576) intitulé "Quel temps font-ils ?" SFRS©.

Le modèle de couleur final contient 9 Gaussiennes. La détection de visages basée sur le modèle de couleur conduit à un ensemble de visage F_{FC} avec 413 visages, avec une précision de 79% (328 régions couvrant de vrais visages ; 85 régions ne couvrant pas de visage), et un rappel de 72%.

L'information apportée par $F_{FC} \cup F_{SVM}$ est de savoir si un plan contient un visage ou non. Sur cette base, 344 plans sur 406 ont été marqués comme contenant au moins un visage (323 plans contenant vraiment un visage ; 21 faux-positifs) pour une précision de 94%.

3.4.0.4 Scènes de dialogue visuelles

Pour la détection de scènes périodiques, la "frame" située au milieu de chaque plan a été extraite et son descripteur MPEG7 [34] "Color Layout Descriptor" a été calculé à l'aide de XM, version 6.0, le logiciel de vérification du standard MPEG7 [16]. Chaque CLD est composé de 6 coefficients pour la luminance et de trois coefficients pour chaque composant de chrominance. La structure MPEG7 "Descriptor Collection Unit" correspondante est stockée au format XML. La matrice de similarité a été calculée en utilisant la formule de associée au CLD [34].

Pour évaluer la qualité des scènes détectées, nous utilisons les mesures de "shot precision" et "shot recall" introduites dans le chapitre 1 (cf. Définition 9) que nous complétons par les mesures de "Scene Precision" et "Scene Recall" :

$$SceneRecision = \frac{\#\ scenes\ correctement\ classifiees}{\#\ scenes\ classifiees} * 100\% \quad (3.20)$$

$$SceneRecall = \frac{\#\ scenes\ correctement\ classifiees}{\#\ scenes\ a\ detecter} * 100\% \quad (3.21)$$

La segmentation de référence des plans en scènes périodiques et en scènes de dialogue visuelles a été faite manuellement pour permettre le calcul des valeurs "ShotRecall" et "ShotPrecision". Ainsi, dans la séquence vidéo traitée, quarante-sept scènes périodiques dont quarante-et-une scènes de dialogue visuelles ont été étiquetées.

Concernant les valeurs de "SceneRecall" et "ScenePrecision", toute scène détectée et incluse dans une scène de référence est considérée comme correcte. Dans le cas d'une sur-segmentation d'une scène de référence couverte par plusieurs scènes détectées, seulement une scène est comptabilisée comme correcte.

Les résultats sont présentés dans le Tableau 3.1. La détection de scènes périodiques conduit à détecter quarante-neuf scènes (38 réelles ; 7 scènes non-périodiques et 4 scènes

	Shot		Scene	
	Recall	Precision	Recall	Precision
Scènes périodiques	58%	79%	81%	78%
Scènes de dialogue visuelles F_{SVM} seul	6%	87%	10%	100%
Scènes de dialogue visuelles $F_{SVM} + F_{FC}$	52%	78%	71%	74%

TAB. 3.1: Rappel et précision de la détection des scènes périodiques et des scènes de dialogue visuelles.

Étape	Temps
Extraction des CLD	1 min
Détection des scènes périodiques	33 sec
Détection de visage par SVM	30 sec
Boucle d'amorçage-filtrage	35 min 16 sec
Détection de visage par modèle de couleur	25 min

TAB. 3.2: Temps de calcul pour chaque étape de la méthode sur un document vidéo de 66 minutes avec 406 (Pentium 4, 2.8 Ghz, 1Go RAM).

sur-segmentées). Trente-huit scènes de dialogue visuelles sont obtenues (29 scènes réelles ; 6 scènes incorrectes et 4 scènes sur-segmentées). Deux raisons expliquent la détection des six scènes incorrectement qualifiées de scènes de dialogue visuelles. D'abord, un certain nombre de parties du corps (mains et bras essentiellement) sont détectés comme visages par le détecteur basé sur la couleur de peau. Cela peut conduire à qualifier une scène périodique comme scène de dialogue à tort. Enfin, des scènes périodiques peuvent impliquer des plans dans les ensembles A et B, qui sont similaires au sens de la mesure de similarité associée au CLD mais pas au sens du jugement humain de la similarité. Ces scènes n'appartiennent donc pas à la segmentation de référence des plans en scènes qui a été effectuée manuellement.

Le filtrage des scènes périodiques ne contenant pas de visages a permis de rejeter dix scènes périodiques. Parmi celles-ci, trois scènes ont été rejetées à tort car les deux détecteurs de visage n'ont pas détecté un ou plusieurs visages pourtant présents dans les plans de ces scènes. Sept scènes périodiques ne correspondant pas à une scène de dialogue ont correctement été éliminées.

Le Tableau 3.1 indique également les résultats obtenus en n'utilisant que les visages issu de F_{SVM}. On remarque les faibles valeurs de rappel, correspondant à la détection de seulement quatre scènes de dialogues visuelles. La détection de visages basée sur la couleur a permis de détecter cent-trente-cinq visages de profil qui n'auraient pas pu être détectées par l'algorithme basé sur SVM. La coopération entre les deux détecteurs est donc très bénéfique.

La méthode proposée est utilisable dans un scénario d'indexation des documents vidéo hors-ligne, car les différentes étapes de la méthode requièrent des calculs intensifs. Le Tableau 3.2 indique les temps de calcul associés aux différentes étapes de la méthode.

3.5 Conclusion et perspectives

Nous avons proposé de caractériser les scènes périodiques d'un document vidéo grâce à la matrice d'adjacence pondérée du "graphe vidéo" (cf. chapitre 2.1.3). Les motifs périodiques correspondant aux scènes périodiques dans le document vidéo sont détectées par seuillage de la matrice et détection de "motifs en damier" à l'aide d'outils morphologiques [141]. Des niveaux de seuillage de plus en plus faibles permettent de détecter des "motifs en damier" imbriqués. La persistance d'un "motif en damier" est mesurée par le nombre de niveaux de seuillage auxquels il existe. Un motif persistant correspond à une scène périodique très contrastée.

L'étape de détection des scènes périodique utilise trois paramètres t_1, t_q et m :

- $t_1 \in [0;1]$ et $t_q \in [0;1]$ désignent respectivement la valeur du niveau de seuillage le plus élevé et le niveau de seuillage le plus faible. Nous préconisons d'utiliser $t_1 = 1$, soit la valeur maximum de la matrice de similarité M et t_q tel que 80% des valeurs de M soient supérieures à t_q.
- le paramètre m permet de régler la persistance minimale d'une scène périodique pour qu'elle soit détectée. La valeur de m représente la longueur d'un sous-intervalle de $[0;1]$. Nous préconisons de fixer m à 20% de l'intervalle.

Nous avons incorporé un indice visuel de haut-niveau dans notre processus de détection. La présence de visages dans tous les plans impliqués dans une scène périodique nous permet de caractériser les "scènes de dialogue visuelles". Pour cela, nous avons proposé une méthode de détection des visages par amorçage-filtrage permettant d'adapter un modèle de couleur de peau aux spécificités d'un document vidéo particulier. Le modèle de couleurs de peau est utilisé pour effectuer une détection de visages selon ce critère. De plus, la boucle d'amorçage-filtrage améliore la précision des résultats issus d'un détecteur de visages par SVM quand la proportion de faux-positifs n'est pas trop élevée.

L'étape d'amorçage-filtrage requiert un paramètre permettant de définir l'intervalle de confiance à utiliser pour conserver ou rejeter un pixel coloré étant donné sa Gaussienne associée. Nous utilisons un intervalle de confiance à 1.5σ.

Chapitre 4

Plongement déterministe d'un espace métrique

Ce chapitre présente un algorithme de MDS (nommé I-Pack, pour "Iterative Packing") [45]. Cet algorithme réalise le plongement en deux dimensions d'un espace métrique donné. La méthode proposée repose sur deux points :

- 1° la construction d'une hiérarchie d'échantillons de plus en plus dense de l'espace métrique. Dans chaque échantillon, toute paire d'éléments appartenant au même niveau d'échantillonage est séparée par une distance minimale (propriété de séparation).
- 2° le placement des points en deux dimensions est tel que la propriété de séparation d'un niveau d'échantillon est respectée en 2D. Plus précisément, dans le contexte de la visualisation d'images, on associe à chaque point un rectangle pouvant contenir l'image associée. Nous désirons un placement des images sans recouvrement.

I-Pack n'est donc pas une méthode basée sur la simulation d'un modèle de force mais un algorithme de plongement basé sur le placement optimal de carrés et sur la construction d'une hiérarchie d'échantillons de l'espace métrique.

Arbre d'échantillonage Récemment, des structures de données efficaces [89, 21, 93, 61] pour l'accès aux plus proches voisins dans un espace métrique (S, δ) ont été proposées. Ces structures sont toutes basées sur une hiérarchie d'échantillons $S = S_0 \subseteq ... \subseteq S_i \subseteq S_{i+1}...S_h$ de l'espace métrique de départ, de laquelle un arbre d'échantillonage T peut être extrait (cf. section 1.5.3). Chaque S_i est une 2^i-séparation et une 2^i-domination de S_{i-1}. Tout élément u de S_{i-1} est donc couvert par au moins un sommet de S_i. Il en choisit un arbitrairement qui sera qualifié de *parent* de u et noté $P(u)$. Ainsi, on peut associer à la hiérarchie d'échantillons un arbre, non défini de manière unique. Le degré de l'arbre d'échantillonage, noté α, dépend du nombre d'échantillons supplémentaires qui peuvent apparaître quand la distance de séparation est divisée par deux. L'importance de cette augmentation dépend de la dimension intrinsèque de l'espace métrique et cette mesure intervient donc dans la complexité en temps des requêtes appliquées à ces structures de données.

Impact de la dimension doublante des données Afin de comparer I-Pack aux méthodes basées sur la simulation d'un modèle de forces et sur l'échantillonnage aléatoire des données, on utilise des jeux de données réels issus de l'indexation automatique de collections d'images, mais également des données aléatoires dont on maîtrise la dimension doublante et l"'aspect ratio". Ces données permettent d'illustrer l'influence de ces para-

mètres sur la qualité des plongements obtenus et de mieux appréhender la nature des données manipulées (dimension intrinsèque, "aspect ratio" et présence de fragments).

Mesure de la qualité du plongement Pour mesurer la distorsion engendrée par un plongement en deux dimensions, deux mesures sont utilisées dans la suite de ce chapitre. D'abord le *stress* (cf.section 1.4), qui est une mesure globale de la qualité du plongement, ensuite *l'étirement* (ou stretch) qui mesure le rapport entre la distance euclidienne et la distance initiale entre deux points.

Plan et résultats Dans la section 4.1, on présente l'algorithme de plongement de l'espace métrique (S, δ) de taille $|S| = n$ et on montre que sa complexité en temps, en $O(n(\alpha \log \alpha) \log A)$, dépend de la dimension intrinsèque de l'espace métrique, exprimée par α et de son "aspect ratio", noté A. Nous présentons dans la section 4.2 une comparaison expérimentale avec d'autres algorithmes de MDS. On montre notamment la rapidité d'exécution d'I-Pack quand la dimension doublante est modérée, ce qui est le cas de nombreux jeux de données réels comme cela a été montré dans [79] : le calcul de la dimension doublante sur une douzaine de jeux de données réels comptant jusqu'à 60.000 éléments conduit à des valeurs de dimension doublante comprises dans l'intervalle $[2; 12]$. Ceci laisse supposer qu'en pratique, les données réelles ont une dimension doublante bornée.

Contrairement aux algorithmes de plongement par modèle de force, I-Pack est déterministe. On montre que grâce au déterminisme de la méthode et à la réservation d'espace dans le plongement on peut positionner de nouveaux éléments sans modifier drastiquement la position des éléments déjà présents. Enfin, on montre que la stratégie d'échantillonage utilisée engendre des valeurs de stress et de stretch meilleurs que les algorithmes de MDS basés sur un échantillonage aléatoire lorsque l'"aspect ratio" des données est grand et qu'elles contiennent de nombreux fragments.

4.1 Plongement I-Pack

Un arbre d'échantillonage T est construit à partir de l'espace métrique (S, δ) associé aux images indexées que l'on veut représenter dans le plan. Pour cela, nous utilisons la méthode présentée en section 1.5. L'algorithme I-Pack décrit dans l'algorithme 4.2 prend comme entrée l'arbre enraciné T.

Nous rappelons que $u^{(i)}$ désigne la copie du sommet u au niveau $0 \leq i \leq \ell_u$. Soit $q_{u^{(i)}}$ le carré associé au sommet $u^{(i)}$ de niveau i. Soit $Q(u^{(i)}) = \{q_v | v \in C_{i-1}(u^{(i)})\}$, l'ensemble de carrés associé aux enfants du sommet $u^{(i)}$. La phase de plongement en 2D consiste à placer chaque image, contenue dans un carré de largeur unitaire, dans une grille discrète.

Le plongement garantit deux propriétés :

- *non-recouvrement* : l'algorithme I-Pack affecte un carré de taille unitaire aux feuilles de T. Chaque sommet interne $u^{(i)}$ dispose les carrés associés aux sommets de $C_{i-1}(u^{(i)})$ sans recouvrement dans son carré $q_{u^{(i)}}$. Cette opération est appliquée récursivement.
- séparation, 2 éléments $u^{(i)}, v^{(i)}$ tels que $P(u^{(i)}) = v^{(i)}$ sont placés à une distance supérieure à 2^i dans la grille.

Pour faciliter la présentation, nous supposons que la distance minimale entre deux éléments de (S, δ) est égale à 1 et qu'il n'existe pas de "doublons" dans S, c'est à dire deux points $u, v \in S$ tels que $\delta(u, v) = 0$. L'algorithme I-Pack peut facilement être modifié pour tenir compte de ces cas de figure.

A partir de l'arbre T, la phase de plongement est basée sur le positionnement suivant : $\forall u^{(i)} \in T, \forall i \leq \ell_u$, on associe à chaque élément de $C_{i-1}(u^{(i)})$ un carré de coté au moins 2^{i-1}, c'est à dire dont la dimension correspond à la séparation au niveau $i-1$. Ainsi, si l'on associe à chaque $v \in C_{i-1}(u^{(i)})$ la position du centre de son carré associé, alors la séparation minimale de 2^{i-1} entre ces points sera garantie en $2D$.

4.1.1 Fonction PACK

La fonction PACK(Q, w) prend en entrée un ensemble de carrés Q et calcule un positionnement sans recouvrement de Q dans un carré de largeur w. Le principe est de positionner les carrés par ordre décroissant de longueur, rangée par rangée, à l'intérieur d'un carré englobant noté q (Voir figure 4.1). Si $|Q| = 1$ alors l'unique carré est centré dans q de coté égal à 2^i.

```
Données: Q ensemble de carrés à positionner, w longueur du
coté du carré final.
abscisse ← 0
ordonnée ← 0
delta ← 0
Trier Q par ordre décroissant de taille
Pour c ∈ Q Faire
    Si abscisse + longueur(c) > w Alors
        ordonnée ← ordonnée + delta
        abscisse ← 0
    FinSi
    Si abscisse = 0 Alors
        delta ← hauteur(c)
    FinSi
    Placer(c, abscisse, ordonnée)
FinPour
```

Algorithme 4.1: Fonction PACK

L'objectif de PACK est de minimiser la surface requise par le carré englobant. Cette méthode de "square-packing" est simple et permet un placement qui occupe une surface qui est une 2-approximation de la surface optimale. Ce résultat est du à Moon et al. [116] qui a déterminé la valeur de w permettant d'obtenir cette approximation.

Théorème 4.1 (Square-packing [116])
Soit Q un ensemble de carrés d'aire totale s, soit w_1 la longueur du coté du plus grand carré de Q. La fonction PACK(Q, w) est une 2-approximation du placement optimal si $w = w_1 + \sqrt{s - w_1^2}$.

La figure 4.2 présente le résultat de l'algorithme pour une collection de 2000 éléments. On voit nettement l'influence de la méthode de "packing" sur le placement des éléments.

4.1.2 Complexité théorique

Proposition 4.2 (Complexité en temps de I-PACK)
Soit T l'arbre de taille N et degré maximum α, associé à l'espace métrique (S, δ). L'algorithme I-PACK *calcule le plongement en 2D de T en temps $O(N\alpha \log \alpha)$.*

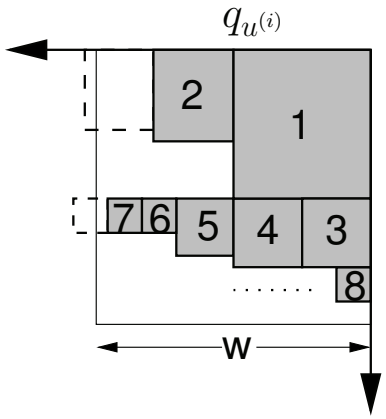


FIG. 4.1: Placement de carrés sans recouvrement. L'ensemble de carrés $Q(u^{(i)})$ (en gris) est placé à l'intérieur du carré $q_{u^{(i)}}$ de largeur w par la fonction PACK.

Données: L'arbre T associé à l'espace métrique (S, δ)
Associer un carré d'aire unitaire, q_u^0, à chaque feuille $u^{(0)} \in T$;
Pour niveau i de 1 à ℓ_v **Faire**
  Pour $v^{(i)} \in T$ **Faire**
    Soit w_1 la largeur du plus grand carré de
      l'ensemble $Q(v^{(i)}) = \{q_{u^{(i-1)}} | u \in C_{i-1}(v^{(i)})\}$;
    Soit s la somme des aires des carrés dans $Q(v^{(i)})$
    Soit $w = \max(2^i, w_1 + \sqrt{s - w_1^2})$;
    Positionner l'ensemble $Q(v^{(i)})$ dans $q_{v^{(i)}}$ en utilisant PACK($Q(v^{(i)}), w$).
  FinPour
FinPour
Associer les coordonnées du centre du carré $q_{u^{(0)}}$ à chaque feuille $u^{(0)} \in T$;

Algorithme 4.2: Algorithme I-PACK

Preuve : La taille de la donnée pour l'algorithme I-PACK est le nombre de sommets de l'arbre T et non le nombre d'éléments de E. Ainsi, il y a $N = O(n \log A)$ sommets dans l'arbre T et autant d'appels à la fonction PACK. En chaque sommet, la fonction PACK doit trier autant de carrés qu'il y a de fils. Le nombre maximum de fils dans T est égal à $\alpha = 2^{O(dd)}$ d'après le Lemme B.1. La complexité en temps de chaque tri est donc en $O(\alpha \log \alpha)$. La complexité en temps totale de I-PACK est donc en $O(N\alpha \log \alpha)$. □

Proposition 4.3 (Complexité en temps du plongement d'un espace métrique par I-PACK)
Soit (S, δ) un espace métrique de dimension doublante dd et d"'aspect ratio" A contenant n éléments. L'algorithme I-PACK *calcule le plongement en 2D de l'espace métrique en temps $O(n2^{O(dd)} dd \log A)$. Si la dimension doublante est bornée, la complexité est $O(n \log A)$.*

Preuve : La méthode se décompose en 2 parties. Le calcul de l'arbre d'échantillonage T à partir de l'espace métrique (S, δ) utilise l'algorithme "Deformable Spanner" qui permet de calculer T en temps $O(n2^{O(dd)} \log A)$ (Lemme 1.3). L'arbre T produit contient $N = O(n \log A)$ sommets et son degré maximum est $\alpha = 2^{O(dd)}$. L'arbre T peut être plongé en

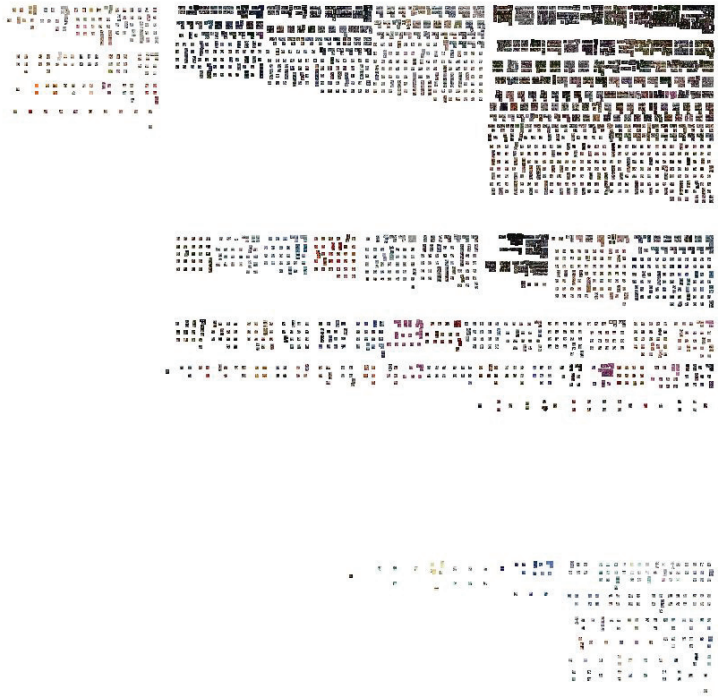


FIG. 4.2: Application d'I-PACK sur un jeu de données composé de 2000 éléments.

2D par l'algorithme I-PACK en temps $O(N\alpha \log \alpha)$ (Lemme 4.2). La complexité totale de la méthode est donc dominée par I-PACK et est en $O(N\alpha \log \alpha)$. □

En pratique, les résultats expérimentaux montrent que le rapport des longueurs et la dimension doublante des données sont bornés par une constante. Ces valeurs sont détaillées dans le Tableau 4.1 et on a $\log_2 A < cte$ et $dd < cte$. Ainsi, la complexité en temps de la méthode proposée est quasi-linéaire en n.

4.2 Résultats expérimentaux

Dans cette section, nous comparons I-PACK à :

- GEM* est le nom que nous donnons à une adaptation de l'algorithme GEM de Arne Frick [57]. Nous avons modifié le paramètre "longueur d'arête désirée", égal à une constante dans la version originale, afin de permettre aux arêtes d'avoir une longueur proportionnelle à la valeur $\delta(u, v)$ associée à l'arête (u, v). Sa complexité en temps est en $O(n^3)$.
- Chalmers03' est un algorithme basé sur la simulation d'un modèle de force [119] utilisant un échantillonage aléatoire des points de l'espace métrique. Sa complexité en temps est en $O(n^{3/2})$.
- Jourdan est un algorithme de MDS similaire à Chalmer'03 [86] mais basé sur un échantillonage différent. Sa complexité est en temps est $O(n \log n)$.

4.2.1 Données utilisées

Afin de comparer I-Pack aux méthodes masse-ressort basées sur l'échantillonnage aléatoire des données, on utilise :

- des jeux de données réels correspondant à des distributions relativement uniformes dans des espaces métriques de dimension comprise entre 3 et 4096.
- Pour simuler des cas extrêmes de distributions non-uniformes, on propose :
  - un jeu de données avec un grand "aspect ratio",
  - un jeu de données aléatoire contenant de nombreux fragments.

Ces données permettent d'illustrer l'influence de ces paramètres sur la qualité des plongements obtenus et de souligner les problèmes liés à l'échantillonnage des données manipulées (dimension intrinsèque, "aspect ratio" et présence de fragments).

Données réelles Le Tableau 4.1 résume les données utilisées pour comparer I-Pack avec les autres algorithmes de MDS. Ces espaces métriques sont composées de n vecteurs numériques avec d composantes munis de la norme Euclidienne.

- Les données SwissRoll sont des coordonnées 3D correspondant à des points échantillonnés sur une surface 2D enroulée (cf. figure 1.29).
- Les données de face sont une collection d'images en niveau de gris de taille 64×64 pixels correspondant à différents rendus d'un modèle de visage 3D dans différentes conditions de pose et d'éclairage.
- Les données Aqua, QTFI et TREC sont composées du descripteur MPEG-7 Color-Layout [34] calculé sur une image-clé extraite des plans des documents vidéo suivant : "Aquaculture en Méditerrannée" ©SFRS, "Quel temps font-ils ?" ©SFRS, le corpus 2002 de TREC Video [123]. (Disponible sur la page Web d'Isomap [152]. Images-clés disponibles sur [123]).

Nom	n	d	$\log_2 \alpha$
SwissRoll	7494	3	3.46
Face	698	4096	5.52
TREC	4000	12	8.59
QTFI	394	12	5.78
Aqua	166	12	4

Tab. 4.1: Données utilisées pour la comparaison des algorithmes de MDS.

Le Lemme B.1 permet de donner une borne inférieure de leur dimension doublante. Sur les jeux de données employés, la dimension doublante est au moins égale 3.

Données avec "aspect ratio" élevé La section 4.2.2, on montre qu' I-Pack fournit un meilleur plongement 2D que les autres algorithmes de MDS considérés pour des données avec un grand "aspect ratio" A. On présente ici la construction d'un jeu de données avec un grand "aspect ratio" A.

Soit $U_m = u_0 \dots u_{m-1}$ une séquence de points avec $\delta(u_j, u_{j+1}) = 2^j$ et $\delta(u_j, u_k) = \sum_{i=j}^{k-1} \delta(u_j, u_{j+1})$. On vérifie que l'"aspect ratio" est 2^{m-1} et qu'il existe un élément par échelle de distance. On considère maintenant l'ensemble de points définis par le produit cartésien U_m^L. L'ensemble U_m^2 peut être vu comme une grille en deux dimensions (cf. figure 4.3).

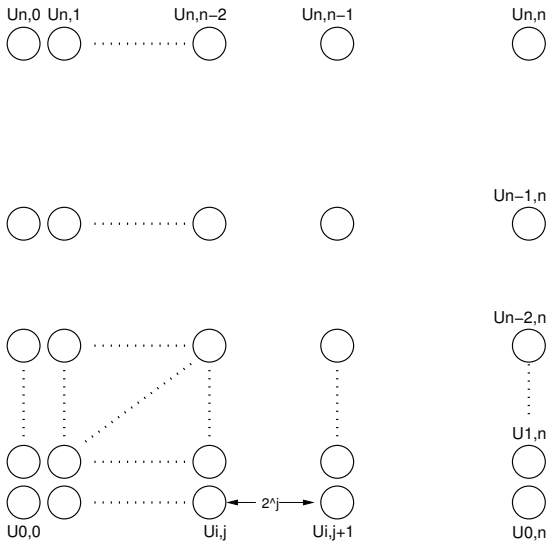


FIG. 4.3: Une grille U_m^2 de taille m^2.

Données avec fragments L'annexe C présente une méthode permettant de générer aléatoirement des données de dimension doublante fixée avec de nombreux fragments. Dans la section 4.2.2, des jeux de données générés avec cette méthode sont utilisés pour la comparaison d' I-PACK et d'autres algorithmes de MDS.

Les dix jeux de données utilisés correspondent à des données aléatoires contenant des fragments de 20 sommets. La distribution de probabilités, notée Π, utilisées pour paramétrer la distribution des degrés de l'arbre aléatoire est $\Pi = (p_0, .., p_i)$ avec $p_i \propto \frac{1}{i^2},\ \forall i > 0$, $p_0 > 0$ et $p_i = 0, \forall i > \alpha$. Cette distribution des degrés correspond à une distribution empirique observée sur des jeux de données réels. La dimension doublante visée est $dd = 5$.

4.2.2 Qualité du plongement

Dans cette partie, nous comparons notre algorithme aux algorithmes de MDS d'après la qualité du placement. D'abord, nous employons la fonction de stress utilisée dans la théorie MDS pour comparer les différents algorithmes, puis nous recourons à la distribution de l'étirement pour donner une comparaison plus détaillée.

Stress Dans [119] et [86], les auteurs évaluent la qualité du plongement 2D produit par leur algorithme basé sur la simulation d'un modèle de force en utilisant le stress de la configuration des éléments en 2D. Intuitivement, plus le stress est bas, meilleure est le plongement puisque les distances originales entre les objets sont bien représentées en 2D. Les valeurs de stress obtenues sont indiquées dans le tableau 4.2. Le meilleur résultat est obtenu avec l'algorithme GEM*, qui produit le plus petit stress à chaque fois. Les autres algorithmes produisent un stress légèrement plus mauvais avec une variation sensible de la valeur de stress au cours de plusieurs essais. L'algorithme I-PACK conduit à des valeurs de stress comparables aux autres algorithmes de MDS.

Étirement Pour une fraction des paires d'éléments, les distances en 2D ne reflètent pas les distances originales. Ceci est mesuré par l'étirement du plongement. Soit (S, δ) un

Layout Algorithm	*Stress* Aqua	*Stress* QTFI	*Stress* TREC
GEM* [57]	0.06	0.08	0.08
Chalmers03' [119]	0.21	0.23	0.19
Jourdan [86]	0.20	0.25	0.20
I-PACK	0.34	0.23	0.23

TAB. 4.2: Valeurs de stress obtenue avec les différents algorithmes de MDS sur les données de Aqua, QTFI and TREC. Les valeurs correspondent à la moyenne des valeurs obtenus pour 10 exécutions sauf pour I-PACK.

espace métrique, soit $\rho : S \rightarrow \mathbb{R}^2$ une fonction de plongement, soit $P_S = \{(x_u, y_u) = \rho(u)\}_{\forall u \in S}$ le plongement des éléments de l'espace métrique (S, δ), soit $|\rho(u) - \rho(v)|$ la distance euclidienne dans le plan 2D entre le plongement de u et celui de v. On définit :

$$Stretch_{uv} = \frac{\delta(u, v)}{|\rho(u) - \rho(v)|} \tag{4.1}$$

La figure 4.4 montre que GEM* conduit à une grande proportion de paires d'éléments avec une valeur d'étirement proche de 1, ce qui indique que la distance dans le plongement est identique à la distance originale entre la paire d'éléments. La distribution de l'étirement associée à l'algorithme Chalmers03' [119] est moins centrée sur la valeur 1 que la distribution associée à GEM* (25.2% ont un étirement > 2 et 0.4% sont $<$ à 0.5). En ce qui concerne la distribution de l'étirement, I-PACK conduit à une tendance comparable avec légèrement moins de valeurs > 2 que Chalmers03' (21.9% ont un étirement > 2 et 3.8% sont < 0.5).

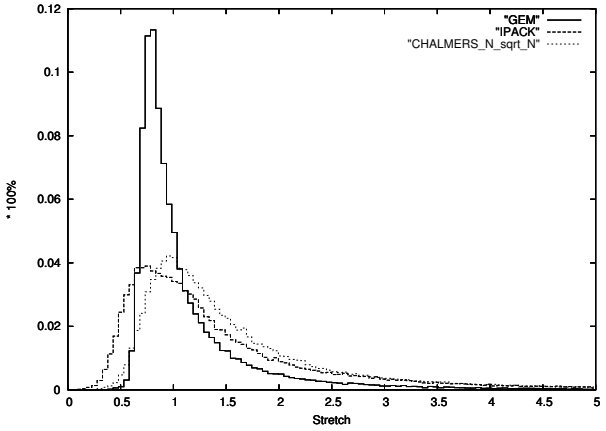


FIG. 4.4: Histogramme de l'étirement entre toute paire d'éléments du jeu de données QTFI.

Données avec un grand "aspect ratio" Le Tableau 4.3 montre les valeurs de stress du plongement de la grille U_{15}^2 décrite dans la section 4.2.1. Les coordonnées des m^2 points de la grille et les distances associées ont été utilisées en entrée de chacun des 4 algorithmes

et les mesures moyennes de stress obtenues sur 10 exécutions du programme sont indiquées dans le Tableau 4.3. Il est clair que d'après la mesure de stress, I-Pack produit un meilleur plongement que les autres algorithmes. I-Pack produit de bons résultats sur des données avec un grand "aspect ratio" en raison de l'échantillonnage hiérarchique des données car le plongement produit par I-Pack garantit la propriété de séparation en 2D.

Layout Algorithm	*Stress*
GEM*	1.90
Chalmers03' [119]	16.36
Jourdan [86]	5.25
I-Pack	0.11

Tab. 4.3: Mesure du stress sur la grille U^2_{15}.

Données contenant des fragments Dans cette partie, nous comparons le niveau de stress obtenus pas les différents algorithmes sur les jeux de données construits selon la méthode décrite dans la partie C.2.

Les résultats du tableau 4.4 indiquent que GEM* et I-Pack produisent toujours un meilleur résultat dans tous les cas avec un léger avantage pour GEM*. Cependant, rappelons que GEM* nécessite plus de temps. Les éléments de fragments proches seront placés dans le carré associé à leur ancêtre commun dans T. Comme l'ancêtre commun de fragments proches est susceptible d'être proche d'eux dans T, les éléments de ces fragments seront placés dans un carré d'aire proportionnelle à leur distance.

Ce placement est optimal en 2D, au sens du stress qu'il produit, dans le cas de α éléments équidistants.

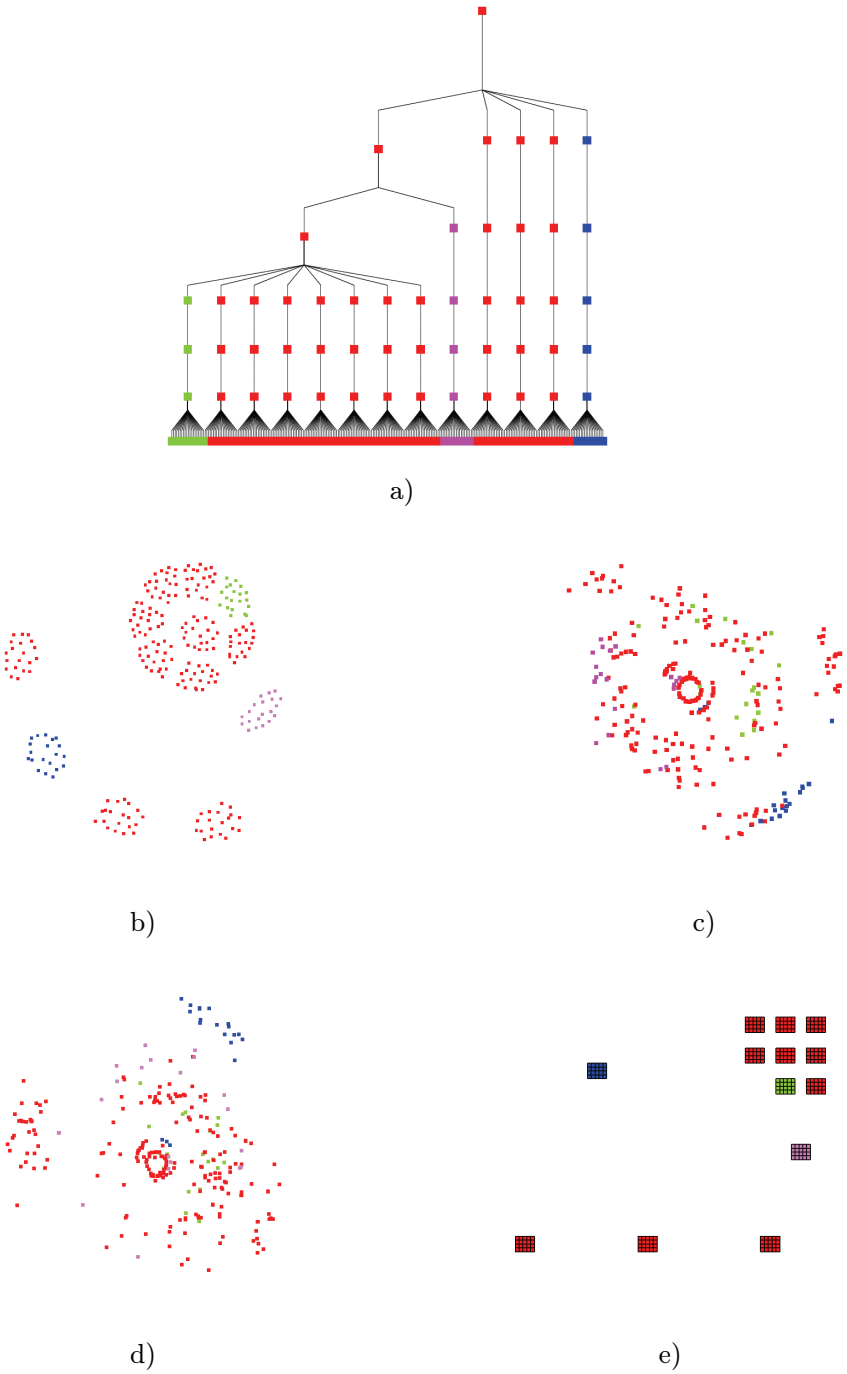
La Figure 4.5 présente les plongements obtenus avec différents algorithmes pour un jeu de données contenant des fragments (cf. Figure 4.5 a)).

4.2.3 Stabilité du plongement

Dans cette partie, on montre qu' I-Pack permet de conserver partiellement le plongement des données lors d'ajouts de nouveaux éléments au jeu de données.

Layout Algorithm	$n = 250$	$n = 500$	$n = 1000$
GEM* [57]	0.42	0.60	0.52
Chalmers03' [119]	7.24	2.21	4.94
Jourdan [86]	9.07	3.69	8.94
I-Pack	0.59	0.99	0.78

Tab. 4.4: Mesure du stress pour des données aléatoires contenant n éléments, avec des fragments de taille $k = 20$ et une dimension doublante $dd = 5$. (Ces jeux de données ont été produits par l'algorithme décrit dans l'annexe C avec le paramétrage $T_{n,20,\Pi}$. La distribution de probabilités Π utilisées pour paramétrer la génération de l'arbre aléatoire sont $\Pi = (p_0, .., p_i)$ avec $p_i \propto \frac{1}{i^2}, \forall i > 0$, $p_0 > 0$ et $p_i = 0, \forall i > \alpha$. 10 jeux de données ont été générés pour chaque valeur de n avec $dd = 5$, ce qui correspond à $\alpha = 32$, et $k = 20$).



a) b) c) d) e)

FIG. 4.5: Comparaison des plongements d'une instance de données aléatoires contenant $n = 260$ éléments, avec des fragments de taille $k = 20$ et une dimension doublante $dd = 5$. a) Arbre généré aléatoirement avec trois fragments distingués (rose, bleu et vert). b) GEM*[57]. c) Chalmers03'[119]. d) Jourdan[86]. e) IPack.

La représentation mentale que chacun se fait des donnée est connue sous le nom de carte mentale. Dans le contexte de la visualisation d'information, on propose à l'utilisateur un plongement des données qui peut constituer une bonne carte mentale pour des données abstraites. Avec I-Pack, on propose une représentation des données basée sur la proximité, dans un plongement en 2D, entre des images dont le contenu est similaire. Dans la partie 4.2.2, nous avons montré l'adéquation entre les distances en 2D et les distances originales. Supposons maintenant que la mesure de dissimilarité utilisée soit en adéquation avec la carte mentale de l'utilisateur, c'est à dire, qu'elle permet à l'utilisateur de s'orienter dans la collection d'images. Il serait alors souhaitable d'obtenir une carte mentale voisine pour la même collection d'éléments à laquelle quelques nouveaux éléments auraient été ajoutés. Nous appelons cette caractéristique la *persistance de la carte mentale* (PCM).

Dans [14], les auteurs introduisent une fonction qui mesure la distance entre 2 dessins de Treemaps. Cette métrique, définie sur l'espace des Treemaps, quantifie les changements dans le dessin lorsque de nouveaux éléments sont ajoutés. Les sommets des Treemaps étant représentés par des rectangles, la métrique mesure la différence entre des rectangles. Soient $r_1 = (x_1, y_1, w_1, h_1)$ et $r_2 = (x_2, y_2, w_2, h_2)$ 2 rectangles définis par les coordonnées de leur coin supérieur-gauche et leurs dimensions, la distance entre r_1 et r_2 est

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (w_1 - w_2)^2 + (h_1 - h_2)^2}.$$

La métrique permettant de comparer 2 instances de Treemaps est la moyenne des distances entre rectangles correspondants.

Afin d'évaluer la PCM d'un algorithme de plongement, on propose d'évaluer le déplacement moyen (DM) d'un ensemble d'éléments de référence, au cours d'exécutions successives d'un algorithme de plongement, appliqué à des collections de plus en plus grandes. Le déplacement moyen mesure les modification d'un plongement existant lors d'ajouts de nouveaux éléments à la collection.

4.2.3.1 Mesure du déplacement moyen

Afin de comparer la PCM associée à I-PACK et celle associée à GEM*, on mesure le déplacement moyen d'un ensemble d'éléments de référence, entre des exécutions d'un algorithme de MDS sur des ensembles de données imbriqués de taille croissante. Le déplacement moyen (DM), mesure les déplacement que subissent les éléments de référence après l'ajout de nouveaux éléments dans le plongement.

Soit l'espace métrique des données (S, δ), soit $E_0 \subset E_1 \subset ... \subset E_k = S$ une séquence de sous-ensembles de S tels que $|E_0| = \frac{n}{2}$ et $|E_{k+1} \backslash E_k| = \sqrt{n}$. Soit P_{E_i}, le plongement de l'ensemble E_i. On définit DM comme la moyenne des déplacements subits par les éléments de E_0 entre P_{E_i} et $P_{E_{i+1}}$ pour un algorithme de plongement donné :

$$DM(P_{E_i}, P_{E_{i+1}}) = \frac{1}{|E_0|} \sum_{u \in E_0} |P_{E_i}(u) - P_{E_{i+1}}(u)|$$

Pour permettre la comparaison des valeurs de DM entre des algorithmes de plongement pouvant conduire à des dessins de dimensions différentes, on normalise les distances dans tous les plongements obtenus avant de calculer DM. Les plongements sont mis à la même échelle de telle manière que la distance maximum entre toute paire de points dans le plongement soit $D_{MAX} = 100$.

4.2.3.2 Comparaison entre GEM* et I-PACK

Le Tableau 4.5 montre la valeur moyenne de DM pour 5 séries aléatoires de sous-ensembles de E.

Exécution	GEM*	I-Pack
1	33.70	10.22
2	40.39	12.21
3	36.76	10.37
4	34.63	5.90
5	36.47	14.31

a) $|E| = 166$

Exécution	GEM*	I-Pack
1	34.01	7.84
2	35.42	9.72
3	36.31	7.84
4	35.62	6.73
5	33.33	8.20

b) $|E| = 394$

Tab. 4.5: Valeur moyenne de DM obtenue pour 5 collections aléatoires. a) Résultats sur les données de Aqua. b) Résultats sur les données de QTFI.

Le tableau 4.5, montre que les valeurs de DM associées à I-Pack sont plus faibles que celles associées à l'algorithme GEM*. I-Pack est donc moins sensible aux ajouts de nouveaux éléments. De plus, pour le même ensemble, plusieurs exécutions d' I-Pack conduisent au même plongement (DM = 0), alors que GEM* produit différents plongements ($DM = 30.23$ en moyenne). L'ajout d'un seul élément peut déranger l'équilibre fragile du modèle de forces utilisé par GEM* qui convergera vers un nouvel équilibre tout à fait différent. I-Pack "gaspille" de l'espace afin de garantir la propriété de séparation mais, il s'avère que cet inconvénient devient un avantage pour la persistance de la carte mentale dans la mesure où l'ajout de nouveaux éléments ne changent pas systématiquement la dimension des carrés englobants et donc le plongement.

4.2.4 Temps d'exécution

Cette section présente la comparaison des temps d'exécution d' I-Pack et des autres algorithmes de MDS pour les mêmes jeux de données. Cela permet de constater qu'en pratique, la complexité en temps d' I-Pack est quasi-linéaire quand la dimension doublante est petite (cf. proposition 1.3).

Une implémentation en C++ de chaque algorithme a été exécutée sur des sous-ensembles du jeu de données SwissRoll sur un Pentium IV 2.8GHz. Le temps d'exécution en secondes est rapporté dans la figure 4.6.

Comme le montre la figure 4.6, I-Pack, Chalmers03' et Jourdan sont beaucoup plus rapides que GEM*. Sur ce jeu de données, I-Pack est environ 3 fois plus rapide que les algorithmes Chalmers03' et Jourdan pour $n = 4000$ et avec des valeurs de stress similaires.

De manière générale et plus théorique, d'après les complexités en temps de la proposition 4.3, I-Pack est plus rapide que tous les autres algorithmes quand $dd = o(\log \log n)$.

4.3 Application à la visualisation de collections d'images indexées

Dans cette section, nous présentons l'application d' I-Pack à une grande collection d'images indexées.

I-Pack a été développé sous forme de plug-in dans la bibliothèque Tulip [6]. Pour la construction de T, nous avons utilisé une implémentation personnelle de "Deformable

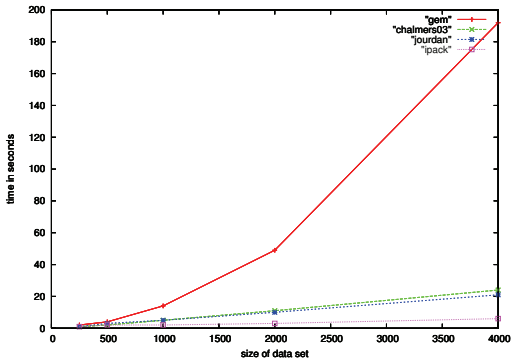
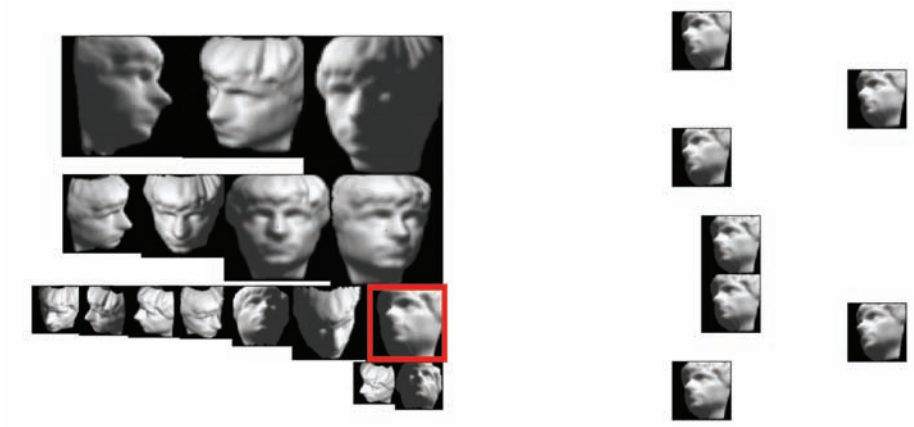


FIG. 4.6: Comparaison des temps d'exécution de I-PACK sur les données SwissRoll.

Spanner" [61] et une implémentation de "Cover Tree" fournie par les auteurs [21]. Avec notre implémentation et le pré-traitement par "Deformable Spanner", nous avons pu afficher les 4000 images-clé du jeu de données TREC.

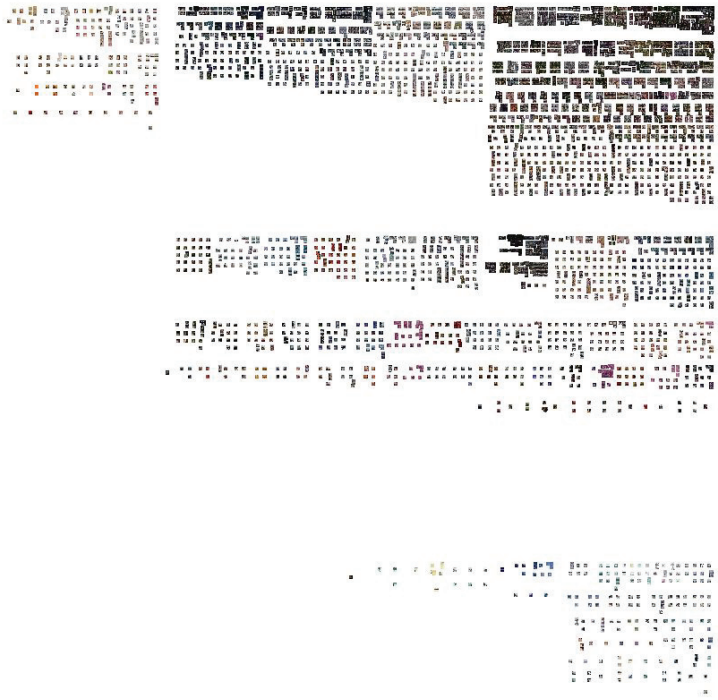
La figure 4.8 montre quelques exemples des résultats obtenus. La vue générale sur la collection d'images entières montre l'influence de la fonction PACK sur le plongement final. L'autre vue montre comment la disposition permet d'identifier des fragments composés d'images-clé semblables au sens de la métrique utilisée (ici la norme Euclidienne sur le descripteur de MPEG-7 ColorLayout).



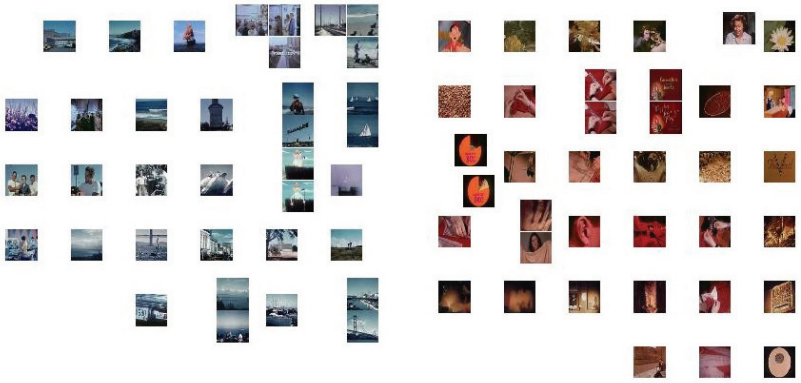
a) b)

FIG. 4.7: Visualisation des images de la collection `face` avec I-PACK. a) Images du niveau S_5. b) Détails d'une zone de S_0 avec des images similaires (pose et illumination similaires). L'ancêtre de niveau 5 de ces images est encadré en rouge dans le niveau S_5.

La figure 4.7 présente des exemples obtenus sur le jeu de données `face`. La figure 4.7 a) montre qu'on peut tirer partie de l'échantillonnage hiérarchique du jeu de données pour ne montrer qu'un niveau de T (ici, les éléments de S_5). La propriété de séparation assure l'affichage d'échantillons représentatifs de la collection entière. La figure 4.7 b) montre un groupe d'images différentes du niveau S_0 partageant un contenu similaire (pose et illumination).



a)



b)

FIG. 4.8: Visualisation de la collection d'images-clé de la collection `trec` avec I-PACK. a) Vue complète. b) Détails d'une zone contenant 2 fragments composés d'images-clé partageant des caractéristiques bas-niveau similaires.

4.4 Conclusion et perspectives

Configuration initiale pour d'autres MDS Le plongement fourni par I-Pack peut servir de configuration initiale pour un algorithme de MDS basé sur un modèle de forces tel que GEM*. La configuration fournie par I-Pack devrait avoir un stress plus faible et permettre à l'algorithme MDS basé sur un modèle de force de converger plus rapidement vers un plongement de stress encore plus faible.

Amélioration de la propriété de séparation La stratégie utilisée par I-Pack ne garantit la propriété de séparation en 2D que pour les sommets de niveau i, cela signifie qu'un certain nombre de relations ne sont pas respectées en 2D. Par exemple, deux sommets appartenant à des sous-arbres différents peuvent se retrouver très proches en 2D car la phase de "packing" associée à leur ancêtre commun peut les placer près de la bordure commune de 2 carrés adjacents.

Chapitre 5

Graphe de navigation et recherche d'images

Dans ce chapitre, nous présentons :
- le concept de *navigation locale* et *graphe navigable* adapté à la recherche recherche dans une collection d'images indexées.
- une construction de graphe navigable (nommé Nav-Graphe).
- une analyse théorique du temps de recherche d'une image. Le temps de recherche est $O(\log A)$ sauts dans le graphe, où A est l'"aspect ratio" de l'espace métrique (S, δ) utilisé.
- une expérimentation qui implémente Nav-Graphe ainsi que les résultats d'un ensemble d'expériences de recherche d'images indexées.
- un descripteur de contenu *interprétable* (nommé descripteur ID pour "Interpretable Descriptor") qui est plus adapté que le descripteur de contenu MPEG-7 "Color Layout Descriptor" dans ce contexte de navigation locale humaine.

État de l'art Les techniques de recherche dans les collections d'images indexées se décomposent en deux catégories :
- Requêtes par l'exemple,
- bouclage de pertinence.

Les techniques de requêtes par l'exemple ("query-by-example", "query-by-feature" et "query-by-sketch") ont été introduites dans QBIC [53] où il est possible de spécifier l'histogramme de couleur d'une image ou d'en produire une représentation schématique à l'aide d'outils de dessin. L'interface RetrievR [98] utilise ce concept dans une interface de dessin ainsi que la technique de "query-by-example" pour la recherche d'images dans la collection FlickR.

Les techniques de bouclages de pertinence ("relevance feedback") permettent de raffiner une requête en sélectionnant des exemples positifs et négatifs à partir d'un échantillon aléatoire ou du résultat d'une recherche par mots-clé. L'ensemble de résultat suivant est choisi de manière à maximiser la similarité avec les exemples positifs et la dissimilarité avec les exemples négatifs, différentes méthodes existent pour cela.

Le système Ikona [26] met en œuvre un système de bouclage de pertinence. Au départ, toute image issue d'un échantillon aléatoire ou de l'ensemble des résultats d'une requête par l'exemple à une pertinence neutre. L'utilisateur forme alors un ensemble d'exemples positifs, noté P, et négatifs, noté N. Le bouclage de pertinence est basé sur le calcul d'une fonction paramétrée par les ensemble P et N. Chaque exemple génère une zone d'influence positive ou négative autour de lui dans l'espace de description. Cette zone est définie par une fonction radiale décroissante, notée F_P pour la zone d'influence positive et F_N pour

la zone d'influence négative. La pertinence d'une image I par rapport aux ensembles P et N est donnée par la fonction de pertinence $J(I) = \sum_{x \in P} F_R(I) - \sum_{x \in N} F_N(I)$. A l'issue de la sélection des ensembles P et N, les images de la collection ayant la meilleure pertinence sont présentée à l'utilisateur et le processus peut être répété autant de fois que nécessaire.

Dans [110], la recherche d'images dans une collection se fait par le parcours d'un arbre dont les sommets représentent les représentants d'une hiérarchie de fragments de l'espace de description obtenue par "k-means". Ainsi, la racine de l'arbre est le représentant de la totalité de la collection et ses fils sont les représentants des fragments obtenus par fragmentation de cet ensemble. La fragmentation est répétée jusqu'à l'obtention du nombre de niveaux hiérarchiques souhaité. La recherche d'une image est effectuée en parcourant la hiérarchie depuis la racine vers les feuilles de l'arbre. L'utilisateur sélectionne un ensemble d'exemples positifs, noté P, parmi les images issues d'un niveau hiérarchique. La matrice de covariance des descripteurs appartenant à l'ensemble P est calculée. Cette matrice est ensuite utilisée pour paramétrer le calcul de la distance (distance de Mahalanobis) entre le descripteur moyen de l'ensemble P et celui des images présentes dans le niveau hiérarchique inférieur. D'après la définition de la distance de Mahalanobis, les descripteurs situés à égale distance d'un descripteur cible ne se situent pas nécessairement à la surface d'une sphère mais à la surface d'une ellipsoïde. La longueur de chaque axe principal de l'ellipsoïde indique le pondération qui est appliquée à la composante correspondante dans le calcul de la distance. Un petit axe indique une faible variance des descripteurs de P selon cet axe et une influence plus importante de cette composante dans le calcul de la distance.

Avantages de la navigation locale La technique que nous proposons dans ce chapitre est basée sur la navigation comme moyen de recherche d'information [46]. Cette approche présente plusieurs avantages par rapport aux approches présentées ci-dessus :

- Les techniques de requêtes par exemple supposent l'existence d'une requête précise, ce qui n'est pas toujours le cas.
- La principale limitation du bouclage de pertinence est liée à la difficulté de trouver des exemples positifs quand l'ensemble d'images constituant la cible recherchée est de petite taille en comparaison avec la collection complète.

Une méthode basée sur la navigation locale dans un graphe autorise une recherche exploratoire qui est utile lorsque la requête ne peut pas être formulée avec précision. Nous proposons une interface de navigation permettant d'effectuer des recherches dans une collection d'images indexées. Les descripteurs extraits des images sont représentés par des vecteurs numériques à D dimensions munis d'une mesure de dissimilarité. Un *graphe de navigation* est construit à partir de l'ensemble des descripteurs. Les sommets du graphe de navigation représentent des images et les arêtes représentent des liens entre images qui sont utiles à la navigation dans la collection d'images. La technique de recherche consiste à effectuer une navigation locale dans ce graphe. L'interface de navigation présente à l'utilisateur un sommet courant dans le graphe de navigation ainsi qu'une représentation du voisinage de ce sommet. La recherche d'une cible (fig. 5.1 en rouge), s'effectue par sauts successifs dans le graphe. La méthode de recherche consiste à choisir un nouveau sommet courant (fig. 5.1 en vert) parmi les sommets voisins (fig. 5.1 en orange). Le critère qui guide le choix du nouveau sommet courant à chaque étape est la similarité avec la cible.

Cette technique de navigation s'apparente au "routage glouton" dans les réseaux, où une succession de décisions basées sur une connaissance locale du réseau permet d'achemi-

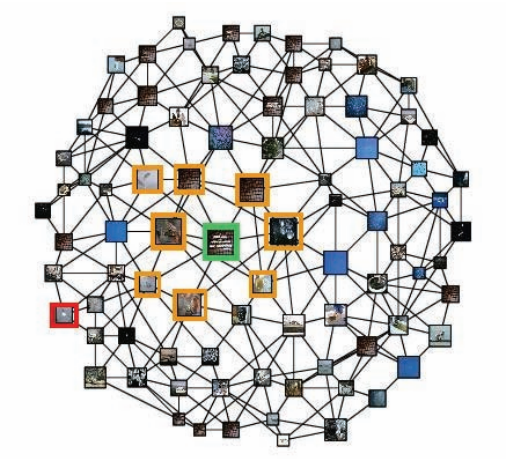


FIG. 5.1: Navigation locale dans un graphe.

ner un paquet d'information vers un nœud précis du réseau. Par "glouton", nous voulons dire qu'aucun mécanisme de retour en arrière n'est nécessaire. Dans notre contexte, le réseau de machines est remplacé par un réseau d'images indexées dont la topologie est conditionnée par les valeurs des descripteurs utilisés et par l'algorithme de construction du graphe de navigation. La destination est constituée par l'image recherchée et l'utilisateur joue le rôle de l'algorithme de routage "glouton". Il choisit, à chaque étape, le sommet voisin du sommet courant qui partage, à son sens, le plus de caractéristiques avec le sommet recherché.

Nous avons identifié des caractéristiques souhaitables concernant la structure de graphe à utiliser pour construire un *graphe de navigation* :

- degré maximum du graphe, indiquant la taille maximale des échantillons d'images présentés à l'utilisateur.
- diamètre du graphe, donnant une indication sur le nombre de sauts minimal à effectuer dans le pire cas.
- présence de "puits" dans le graphe : un puits est un sommet u tel que tous ses voisins sont plus dissimilaires de la cible que u.

Ces caractéristiques de graphe ont une influence en termes de performances sur l'algorithme de navigation locale utilisé ainsi que sur l'efficacité de l'interface de navigation. De manière générale, nous utilisons le terme de *graphe navigable* si le graphe ne possède aucun puits.

Dans ce chapitre nous allons répondre à plusieurs questions essentielles pour la mise en œuvre pratique d'une interface de recherche basée sur un *graphe de navigation* :

- Dans la section 5.1, nous discutons des caractéristiques souhaitables pour un graphe de navigation.
- Dans la section 5.2 nous présentons la construction d'un NAV-GRAPHE.
- Dans la section 5.3 nous présentons les expériences réalisées. Nous analysons les caractéristiques des descripteurs d'images les plus adaptés à la navigation que nous proposons.
- La section 5.4 présente les résultats expérimentaux issus de sept expériences de recherche d'images conduites sur onze utilisateurs-test.

5.1 Graphes et routage "glouton"

Nous proposons d'aborder la recherche d'informations stockées dans les sommets d'un graphe comme un processus de routage "glouton" et donc basé uniquement sur une connaissance partielle et locale du graphe dans son ensemble. Dans la section 5.1.1, nous présentons trois algorithmes de recherche dans les graphes. Nous présentons ensuite trois structures de graphes ainsi que leurs caractéristiques dans le contexte du routage "glouton".

5.1.1 Algorithme de navigation

Nous présentons trois algorithmes de navigation locale pour la recherche d'un sommet cible t en partant d'un sommet source u dans un graphe $G = (V, E)$. Les sommets de G appartiennent à un espace métrique (S, δ) (i.e. $V \subseteq S$).

Soit un graphe $G = (V, E)$, soit un ensemble de sommets $M \subseteq V$, on note $vois(M)$ l'union des voisins des sommets de M :

$$vois(M) = \bigcup_{v \in M} vois(v).$$

D'un point de vue pratique, si un utilisateur suit l'un des algorithmes suivants, le processus peut être décrit ainsi :

1. On présente un échantillon $X \subset S$ d'images. X est soit le voisinage du sommet courant (algorithmes `glouton+` et `glouton`) ou bien le voisinage des sommets déjà visités (algorithme `generique`).
2. L'utilisateur choisit un élément v de X qui lui semble plus similaire à la cible que l'image courante u (i.e. $\delta(v, t) < \delta(u, t)$).

Mesure de performance : pour mesurer la performance du couple (graphe, algorithme de navigation), on considère les critères importants du point de vue de l'utilisateur :
- le temps de recherche, modélisé par le nombre de sauts qui est égal à $|M|$.
- la "complexité" de la recherche, modélisée par la taille des échantillons qui sont proposés, notée $|X|$.

Dans les algorithmes suivants, M désigne l'ensemble des sommets visités, c'est-à-dire, la mémoire de l'utilisateur. La notation, $X \backslash M$ représente l'ensemble X privé de M.

Données: G, source u, cible t
Ensemble M $\leftarrow u$
$v_{min} \leftarrow$ l'élément de $vois(M) \backslash M$ le plus similaire à t
TantQue $u \neq t$ **Faire**
    $u \leftarrow v_{min}$
    Ajouter(u,M)
    $v_{min} \leftarrow$ l'élément de $vois(M) \backslash M$ le plus similaire à t
FinTantQue

Algorithme 5.1: Algorithme `generique`

La dissimilarité $\delta(u, v)$ donnée par la fonction de distance dans l'espace métrique ne doit pas être confondue avec $d_G(u, v)$ qui est la distance dans le graphe. On suppose que pour u et v donnés, on peut calculer $\delta(u, v)$.

Données: G, source u, cible t
Ensemble M $\leftarrow u$
$v_{min} \leftarrow$ un élément de $vois(u) \backslash M$ plus similaire à t que u
TantQue $u \neq t$ **Et** $v_{min} \neq$ **Null Faire**
    $u \leftarrow v_{min}$
    Ajouter(v_{min},M)
    $v_{min} \leftarrow$ un élément de $vois(u) \backslash M$ plus similaire à t que u
FinTantQue

Algorithme 5.2: Algorithme `glouton`

Données: G, source u, cible t
Ensemble M $\leftarrow u$
$v_{min} \leftarrow$ l'élément de $vois(u) \backslash M$ le plus similaire à t
TantQue $u \neq t$ **Et** $v_{min} \neq$ **Null Faire**
    $u \leftarrow v_{min}$
    Ajouter(v_{min},M)
    $v_{min} \leftarrow$ l'élément de $vois(u) \backslash M$ le plus similaire à t
FinTantQue

Algorithme 5.3: Algorithme `glouton+`

Les algorithmes `generique`, `glouton+` et `glouton` sont trois algorithmes de recherche basés sur la navigation dans G. Dans les trois cas, les sommets visités sont mémorisés (dans l'ensemble M) et un sommet déjà visité ne peut être visité de nouveau.

L'algorithme `generique` trouvera la cible quelque soit le graphe utilisé. Il s'agit d'une version de parcours en largeur en partant d'une source,"dirigé" vers la cible.

Les algorithmes `glouton` et `glouton+` sont plus "exigeants" quant à la structure du graphe sur lequel ils s'exécutent. Si le sommet courant ne dispose pas de voisins pouvant faire diminuer la distance à la cible, ces deux algorithmes s'arrêtent. Ils ne permettent d'atteindre la cible que si le graphe est navigable.

En présence d'un puits, l'algorithme `generique` utilise sa mémoire pour trouver le prochain sommet à visiter à partir de son "historique de navigation". Dans les cas favorables, seulement quelques sauts ne rapprochant pas de la cible sont effectués avant de sortir des puits. Dans le pire cas, la totalité du graphe doit être explorée avant de trouver la cible.

La *présence de puits* est donc un critère important car il va conditionner la quantité de mémoire à utiliser dans le cas de l'algorithme `generique`. Dans le cas des algorithmes `gloutons`, un puits peut conduire à l'échec de la recherche.

Considérons maintenant des graphes navigables. Un exemple trivial de graphe navigable est le graphe complet. Chaque sommet est relié à tous les autres sommets et possède une connaissance de l'ensemble des sommets. La quantité de mémoire requise pour stocker le graphe est en $O(n^2)$, le temps de recherche est $|M| = 1$ et la complexité de la recherche est linéaire $|X| = O(n)$.

Pour la navigation, on souhaite idéalement obtenir un *graphe peu dense* dans lequel le routage glouton puisse s'effectuer rapidement. Le degré moyen du graphe est un bon indicateur de sa densité en arêtes et constitue une caractéristique concernant l'efficacité du stockage du graphe de navigation.

Enfin, la caractéristique du graphe qui influence la complexité en temps de l'algorithme de routage est le *diamètre* du graphe. Si le diamètre du graphe est grand, alors le nombre de sauts maximum requis, $|M|$, pour trouver une cible sera de l'ordre du diamètre, c'est-à-dire $|M| \geq diam(G)$. Dans le cas d'un graphe formé d'une chaîne de sommets, le diamètre sera linéaire et le temps de recherche également.

Voyons maintenant des algorithmes de construction de graphes de navigation adaptés à notre contexte et leurs caractéristiques en termes de : degré maximum, densité, diamètre et présence de puits.

5.1.2 Routage dans les graphes géométriques

Intéressons-nous à des graphes navigables dont les sommets dont des points dans $\mathbb{R}^d$. Un *graphe géométrique* est un graphe dessiné dans le plan dont les sommets sont des points et dont les arêtes sont représentées par des segments de droites. Cette notion peut être étendue a des espaces de dimension supérieure.

Les descripteurs associés aux images indexées peuvent être considérés comme des points dans un espace à d dimensions. A partir d'un ensemble de n points, on souhaite trouver un ensemble d'arêtes qui conduit à un graphe géométrique "efficace".

De nombreux travaux concernent la construction de graphes géométriques peu denses en dimension deux. Les *graphes de Yao* ou $\theta-$graphes [175] sont des exemples de graphes géométriques.

Définition 61 (Secteur angulaire)
Soient $u, v \in \mathbb{R}^2$, soit k le nombre de secteurs angulaires d'angle $\theta = \frac{2\pi}{k}$

$$sec(u, v) = \lfloor \frac{\widehat{uv}}{\theta} \rfloor \tag{5.1}$$

désigne l'indice du secteur angulaire de u dans lequel se trouve v.

Définition 62 (Graphe de Yao)
Soit $V \subset \mathbb{R}^2$, soit δ une distance dans $\mathbb{R}^2$. Le graphe de Yao, ou $\theta-$graphe est le graphe orienté $G = (V, E)$ défini par l'ensemble des arêtes

$$E = \{(u, v) | \forall w \neq u : sec(u, v) = sec(u, w) \implies \delta(u, v) < \delta(u, w)\}. \tag{5.2}$$

La construction du graphe de Yao repose sur la création de secteurs angulaires autour de chaque point de l'espace et sur l'ajout d'une arête entre le point courant et son plus proche voisin dans chaque secteur angulaire. Des variantes de cette construction permettent de réduire le nombre d'arêtes ou de borner le degré entrant de chaque sommet : "Symmetric Yao", "Sparse Yao" et "Bounded Yao". Ces constructions sont efficaces pour la création de graphes géométriques connexes dans lesquels le *routage glouton peut s'effectuer sans mémoire* : on peut donc utiliser les algorithmes `glouton+` et `glouton` efficacement avec les graphes de Yao.

Cependant, quand la dimension de l'espace augmente le degrés sortant des sommets croit de manière exponentielle. La construction , en dimension supérieure, est similaire à la construction en 2D à la différence que les secteurs angulaires sont remplacés par des cônes alignés avec les diagonales d'un cube en dimension d [175]. L'inconvénient est que le degré sortant maximum de chaque sommet est $O(2^d)$.

Ce phénomène se retrouve dans d'autres constructions. La triangulation de Delaunay en dimension deux a $O(n)$ arêtes, ce nombre passe à $O(n^{4/3})$ en dimension trois et à $O(n^2)$ en dimension supérieure à trois.

Si le descripteur utilisé est de petite dimension, alors les graphes de Yao sont intéressants car $|X| = O(2^d)$.

5.1.3 Routage dans les graphes des k plus proches voisins

Dans [2], les auteurs proposent de construire un index distribué pour l'organisation d'une collection d'images indexées par des descripteurs de couleur ISO/MPEG-7.

La structure de données, nommée SWIM, est un graphe de navigation dont les sommets correspondent aux images d'une collection. Chaque image établit un lien les k images de la collection les plus similaires selon le descripteur utilisé. Comme une telle construction ne peut pas garantir que le graphe soit connexe, une structure d'anneau est ajoutée au graphe pour garantir cette propriété indispensable au routage.

Dans le pire cas, la recherche d'une image est effectuée par un agent qui implémente un algorithme de routage glouton avec mémoire dans ce graphe (essentiellement l'algorithme `generique`). Les résultats obtenus montrent un nombre de sauts $|M|$ en $O(n)$ pour effectuer une recherche dans la collection d'images. Un intérêt de cette méthode est son indépendance vis-à-vis du nombre de dimensions associées au descripteur. En revanche, le routage ne peut pas se faire sans mémoire car la structure de graphe peut conduire à l'existence de "puits".

5.1.4 Routage dans les arbres

La recherche d'information dans un index représenté par un arbre [68, 99] correspond au parcours de la racine vers le sommet contenant l'information recherchée. Dans les cas favorables, ce nombre est égal à la hauteur de l'arbre et correspond à un nombre de sauts logarithmique en fonction du nombre de sommets de l'arbre, pour un arbre équilibré. Toutefois, dans le contexte de la recherche visuelle, une erreur peut être commise lors du processus de recherche. Cela peut conduire à l'exploration complète d'un sous-arbre ne contenant pas l'information souhaitée avant de conclure que l'information recherchée ne s'y trouve pas. La structure d'arbre seule n'est donc pas adaptée au contexte de la recherche visuelle où les erreurs d'interprétation peuvent être fréquentes.

Pour qu'un arbre de recherche de type kd-tree [99] (pour "k-dimensional tree") soit navigable, il faut choisir $k = 2^d$.

5.1.5 Synthèse

Le tableau 5.1 résume les caractéristiques des structures de graphes de navigation envisageables. Pour les graphes navigables, nous considérons que l'algorithme de recherche est `glouton+`, sinon l'algorithme `generique` est utilisé.

Un compromis intéressant consisterait à obtenir un faible diamètre (logarithmique) ainsi que l'absence de puits pour permettre un routage glouton sans mémoire. Un degré maximum constant serait un avantage supplémentaire, surtout dans le cadre de la navigation humaine.

5.2 Graphe de navigation hiérarchique

Dans cette section, nous définissons la structure de graphe de navigation hiérarchique utilisée pour représenter une collection d'images indexées. Nous partons d'un arbre d'échantillonage T de l'espace métrique (S, δ) pour construire le graphe de navigation hiérarchique (nommé NAV-GRAPHE), noté $G_{T,c}$. Dans la suite de cette section, on supposera

Graphe	Degré sortant max.	Diamètre	Navigable	Densité
Graphe complet	n	1	oui	forte ($\|E\| = n^2$)
Arbres $k-$aire	k	$O(\log_k n)$	oui si $k \geq 2^d$, sinon non **	moyenne ($\|E\| = k.n$)
Yao	$O(2^d)$	$O(n^{1/d})$ *	oui	dépend de d et n
Swim	k	-	non	moyenne ($\|E\| = k.n$)
NAV-GRAPHE	$2^{O(dd)}$	$O(\log A)$	oui	$2^{O(dd)}n$

TAB. 5.1: Comparaison des constructions de graphes navigables envisageables. * si la distribution des points dans $\mathbb{R}^d$ est uniforme. ** en pratique $dd \ll d$.

que $\forall u, v \in S, \delta(u, v) \geq 1$. Nous analysons ensuite la définition d'un paramètre important pour la navigation : la valeur du rayon définissant le voisinage de chaque sommet dans le NAV-GRAPHE.

5.2.1 Définition

Le NAV-GRAPHE $G_{T,c}$ est défini à partir d'un arbre d'échantillonage T (cf. définition 49).

Définition 63 (Arêtes de voisinage)
Soit (S, δ) un espace métrique, soit $H_S = \{S_i\}_{i\in[1,h]}$ une hiérarchie des centres discrets de S et soit $c \in \mathbb{R}^+$ une constante. On définit l'ensemble des arêtes de voisinage *d'un point $u \in S_i$ dans un niveau de la hiérarchie des centres discrets, noté $E_i(c)$, par :*

$$E_i(c) = \{(u, v) \in S_i | \delta(u, v) \leq c.2^i\}. \tag{5.3}$$

La constante c est nommée coefficient de voisinage.

Définition 64 (NAV-GRAPHE de (S, δ))
Soit (S, δ) un espace métrique, soit T un arbre d'échantillonage et soit c un coefficient de voisinage. Le NAV-GRAPHE *de (S, δ) est le graphe $G_{T,c}$ défini par :*

$$V(G_{T,c}) = V(T) \tag{5.4}$$

$$E(G_{T,c}) = E(T) \cup \{E_i(c)\}_{i\in[1,h(T)]}. \tag{5.5}$$

D'après la définition des centres discrets et de l'arbre d'échantillonage (cf. définitions 48 et 49), on peut construire différents arbres d'échantillonage à partir d'un ensemble de centres discrets. De même, il peut exister plusieurs hiérarchies de centres discrets associées à un espace métrique (S, δ). Le NAV-GRAPHE $G_{T,c}$ n'est donc pas unique.

Notons $N_i(u)$ l'ensemble des voisins du sommet $u \in S_i$ dans S_i :

$$N_i(u) = \{v \in S_i | (u, v) \in E_i(c)\}.$$

Proposition 5.1 (Caractéristiques du NAV-GRAPHE)
Soit $G_{T,c}$ un NAV-GRAPHE *défini à partir d'un espace métrique (S, δ) de dimension doublante dd et d"'aspect ratio" A. On a :*

- $|V(G_{T,c})| = O(n \log A)$,
- $|N_u(c)| \leq (4c)^{dd},\ \forall u \in G_{T,c},\ c \in \mathbb{R}^+$,
- $|E(G_{T,c})| = O(nc^{dd} \log A)$.

Preuve de la proposition 5.1 : • Montrons $|V(G_{T,c})| = O(n \log A)$: La hauteur de T, $h(T)$ est égale à $\lceil \log A \rceil$. Le nombre de feuilles de T est égal à n. On a donc $|V(G_{T,c})| = O(n \log A)$.

• Montrons $|N_u(c)| \leq (4c)^{dd}$: comme (S, δ) est de dimension doublante dd alors $B_u(c.2^i)$ est couverte par au plus 2^{dd} boules de rayon $\frac{c}{2}.2^i$. Chacune de ces boules peut elle même être couverte par 2^{dd} boules de rayon $\frac{c}{4}.2^i$. En itérant, on obtient que $B_u(c.2^i)$ peut être couverte par $2^{dd.(log_2(c)+2)}$ boules de rayon 2^{i-2}.

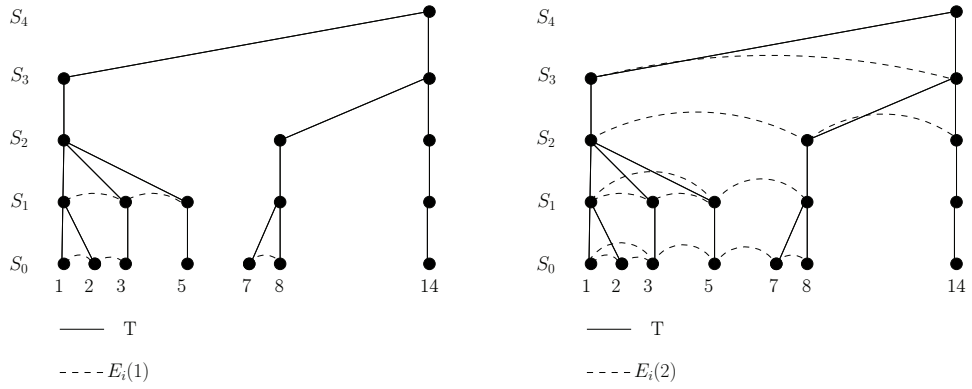
S_i étant une 2^i-séparation, toute boule de rayon 2^{i-2} ne peut contenir qu'au plus 1 élément de S_i. $B_u(c.2^i)$ contient donc au plus $(4c)^{dd}$ éléments de S_i.

• Montrons $|E(G_{T,c})| = O(nc^{dd} \log A)$: On a $|V(G_{T,c})| = O(n \log A)$ et $|N_u(c)| \leq (4c)^{dd}$, le nombre d'arêtes est donc $O(nc^{dd} \log A)$. □

Adaptation des algorithmes de routage On doit modifier les algorithmes `glouton` et `glouton+` pour les appliquer à un NAV-GRAPHE. En effet, il existe une configuration posant problème : si le sommet courant $u \in S_i$ est le sommet courant le plus proche de t, $\ell_u > i$ et u possède deux voisins non visités qui sont $u^{(i+1)}$ et $u^{(i-1)}$ tels que $\delta(u,t) = \delta(u^{(i+1)},t) = \delta(u^{(i-1)},t)$ alors l'algorithme `glouton` (ou `glouton+`) peut choisir indifféremment $u^{(i+1)}$ ou $u^{(i-1)}$ comme nouveau sommet courant. Or, l'un de ces deux sommets se trouve au niveau supérieur et l'autre au niveau inférieur de celui de u. On considérera que dans ce cas de figure, c'est $u^{(i-1)}$, le *voisin de niveau le plus bas qui est choisi.*

5.2.2 Paramétrage du coefficient de voisinage

Nous analysons l'influence du coefficient de voisinage c qui conditionne l'existence des arêtes entre sommets du même niveau de la hiérarchie des centres discrets. Ce paramètre est essentiel pour les performances de la navigation. Nous montrons que sa valeur théorique optimale est égale à 6. Cette valeur permet d'atteindre tout élément de S_0 situé à distance maximale 2^{i+1} du sommet courant sans avoir à remonter dans les niveaux de la hiérarchie lors de la navigation.



a) $G_{T,1}$. b) $G_{T,2}$.

FIG. 5.2: Influence du coefficient de voisinage sur la navigation. Nav-graphe $G_{T,c}$ avec $S = \{1, 2, 3, 4, 5, 7, 8, 14\}$ et $\forall x, y, \in S, \delta(x, y) = |x - y|$.

Le graphe de la figure 5.2.a) correspond à l'utilisation d'un coefficient de voisinage égal à 1. On souhaite trouver la cible constituée par le sommet $7^{(0)}$ ($u^{(i)}$ désigne la copie du sommet u au niveau S_i) en partant de la racine du NAV-GRAPHE (copie du sommet 14 au niveau 4, noté $14^{(4)}$). L'algorithme `glouton+` (ou `glouton`) utilisé pour rechercher la cible constituée par le sommet $7^{(0)}$ depuis la racine du NAV-GRAPHE va effectuer le parcours de la suite de sommets suivants :

$$14^{(4)}1^{(3)}1^{(2)}5^{(1)}5^{(0)}.$$

La navigation conduit dans un puits et l'algorithme `glouton+` (ou `glouton`) ne peut plus trouver de sommet non visité qui rapproche de la cible. Pour sortir de cette situation, une stratégie différente doit être utilisée. Par exemple, effectuer un parcours en largeur du graphe à partir du sommet $5^{(0)}$ jusqu'à ce qu'un sommet qui rapproche de la cible soit trouvé (le sommet $8^{(2)}$ dans ce cas). On doit donc visiter la totalité du sous-arbre $T_{1^{(3)}}$, ce qui est coûteux en temps et en mémoire.

Le graphe de la figure 5.2.b) correspond à l'utilisation d'un coefficient de voisinage égal à 2. L'algorithme `glouton+` (ou `glouton`) pour rechercher le sommet $7^{(0)}$ va effectuer le parcours de la suite de sommets suivante :

$$14^{(4)}1^{(3)}1^{(2)}8^{(2)}8^{(1)}7^{(0)}.$$

La présence d'une arête de voisinage entre le sommet $1^{(2)}$ et le sommet $8^{(2)}$ permet d'éviter la situation précédente. Un paramétrage trop restrictif du coefficient de voisinage peut conduire à l'exploration inutile d'un sous-arbre entier. Un paramétrage trop important peut faire croître le degré de manière drastique, ce qui n'est pas souhaitable. La proposition 5.2 présente un paramétrage théorique optimal pour c.

Proposition 5.2 (Coefficient de voisinage optimal)
Pour toute cible, l'algorithme `glouton+` *arrive a destination dans $G_{T,c}$ en $O(\log A)$ sauts si et seulement si $c \geq 6$.*

Preuve de la proposition 5.2 : Supposons que l'algorithme `glouton+` fonctionne avec un bon paramétrage de c. Rappelons que $\forall i \in [1,h]$, S_i est une $2^{i+1}-$domination de S (cf. lemme 1.2), donc $\forall t, \exists u \in S_i$ tel que $\delta(u,t) \leq 2^{i+1}$.

Supposons que le sommet courant $u \in S_i$ soit à distance inférieure à 2^{i+1} de t. Montrons qu'en au plus 3 sauts, on atteint un sommet de S_{i-1} à distance inférieure à 2^i de t.

• $Cas\ 1$: $v^{(i)}$, le sommet de S_i le plus proche de t est différent de u. On veut pouvoir l'atteindre en 1 saut donc $v \in N_u(c)$. Sachant que $\delta(v^{(i)},t) < \delta(u,t) \leq 2^{i+1}$, on a $\delta(v^{(i)},u) \leq 2^{i+2}$. Un coefficient de voisinage $c \geq 4$ permet de donc de trouver ce plus proche sommet. Passons maintenant au cas 2 en prenant $u \leftarrow v^{(i)}$.

• $Cas\ 2$: $u = u^{(i)}$ est le sommet de S_i le plus proche de t. On applique donc $u \leftarrow u^{(i-1)}$. Nous avons 2 possibilités :

- Si $\delta(u,t) \leq 2^i$; nous avons vérifié l'induction.
- Si $2^{i+1} \geq \delta(u,t) > 2^i$ alors il existe $v \in S_{i-1}$ tel que $\delta(v,t) \leq 2^i$. Selon l'hypothèse que l'on a un bon paramétrage de c, ce sommet v doit être atteignable en un saut, donc $v \in N_u(c)$. Nous avons donc $\delta(u,v) \leq \delta(u,t) + \delta(t,v) \leq 2^{i+1} + 2^i = 6.2^i - 1$. La valeur du coefficient de voisinage $c \geq 6$ permet donc de trouver le sommet le plus proche.

La distance à la cible t est donc divisée par 2 en au plus 3 sauts et on passe d'un niveau i à $i-1$ en au plus 2 sauts. En partant du niveau $h = \lceil \log A \rceil$, l'algorithme `glouton+` permet d'atteindre la cible en au plus $2h$ sauts en prenant $c = 6$.

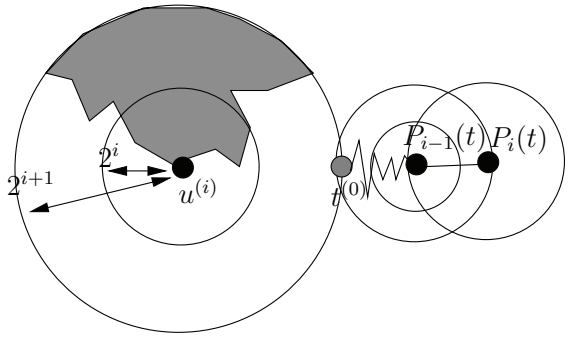


FIG. 5.3: Définition du voisinage de niveau i.

Peut-on prendre $c < 6$? Le fait que S_{i-1} soit une 2^i-domination de S_0 ne garantit l'existence que d'un seul sommet à distance inférieure ou égale à 2^i pour chaque cible. Si $u \in S_{i-1}$ est le sommet courant à distance $2^i < \delta(u,t) \leq 2^{i+1}$, alors il faut que $N_u(c)$ contienne ce sommet pour pouvoir se rapprocher de la cible au niveau $i-1$. Si on décide de descendre de niveau, c'est-à-dire, aller au niveau $i-2$, on ne fait que décaler le problème dans les niveaux inférieurs et la valeur de c nécessaire devra être plus élevée dans les niveaux suivants. □

Proposition 5.3 (Temps d'une recherche avec l'algorithme `glouton`)
L'algorithme ***`glouton`*** *trouve une cible dans $G_{T,c}$ en $O(2^{O(dd)} \log A)$.*

Preuve : Le schéma de preuve est le même que pour la preuve de la proposition 5.2. L'argument qui change concerne le nombre de sauts maximal à effectuer pour atteindre le sommet de S_i le plus proche de la cible t, nommé $v^{(i)}$.

Supposons que $u \in S_i$ soit le sommet courant à distance inférieure à 2^{i+1} de t. Le nombre de sommets de S_i pouvant potentiellement rapprocher de $v^{(i)}$ est $|B_t(2^{i+1}) \cap S_i| = |X| = k$. $B_t(2^{i+1})$ peut être couvert par au plus 2^{dd} boules de rayon 2^i. Si l'on itère deux fois cet argument, $B_t(2^{i+1})$ peut être couvert par au plus 2^{3dd} boules de rayon 2^{i-2}. Chacune d'elle contient au plus un élément de S_i donc $k \leq 2^{3dd}$.

Dans le pire des cas, on peut parcourir tous les éléments de X en se rapprochant de $v^{(i)}$ à chaque saut. Le temps de recherche pour l'algorithme `glouton` est donc en $O(2^{3dd} \log A)$.□

5.3 Expérimentations

Dans cette section, nous présentons l'interface de navigation basée sur le NAV-GRAPHE $G_{T,c}$ associé à une collection d'images indexées.

Nous présentons les collection d'images ainsi que les descripteurs utilisés pour construire les graphe de navigation hiérarchique associé. Nous utilisons le descripteur ISO/MPEG-7 "Color Layout Descripteur" et nous proposons un descripteur d'images constitué d'indices visuels interprétables, nommé ID pour "Interpretable Descriptor".

Nous décrivons ensuite les expériences utilisées pour :

- comparer l'interface de navigation hiérarchique avec la recherche linéaire,
- mesurer l'influence du coefficient de voisinage sur les performances des recherches,
- comparer les descripteurs CLD et ID dans le cadre de la navigation locale.

5.3.1 Interfaces utilisateur

Cette sous-section présente les deux types d'interfaces utilisées pour effectuer les expérimentations. L'interface de recherche hiérarchique permet d'effectuer la navigation dans le graphe de navigation hiérarchique, $G_{T,c}$, associé à une collection d'images indexées. L'interface linéaire permet d'effectuer une recherche dans une liste des éléments de S.

5.3.2 Recherche hiérarchique

La figure 5.4 présente l'interface utilisateur permettant de naviguer dans une collection d'images indexées. Le sommet courant u est affiché en première position et encadré en rouge. Le voisinage de ce sommet $N_u(c)$ est affiché après et les sommets sont classés par ordre décroissant de similarité avec le sommet courant. Celui-ci peut être changé en sélectionnant un des voisins du niveau courant.

Dans le cadre d'une tâche de recherche dans la collection d'images, l'image cible est affichée en permanence en haut et à droite de l'écran. Le point d'exclamation rouge permet d'abandonner la recherche en cours.

Si la vue locale présente un nombre d'éléments trop important, les barres de défilement de l'ascenseur du navigateur permettent de faire défiler la vue locale. Dans ce cas, le menu de navigation et la cible suivent le défilement de l'ascenseur.

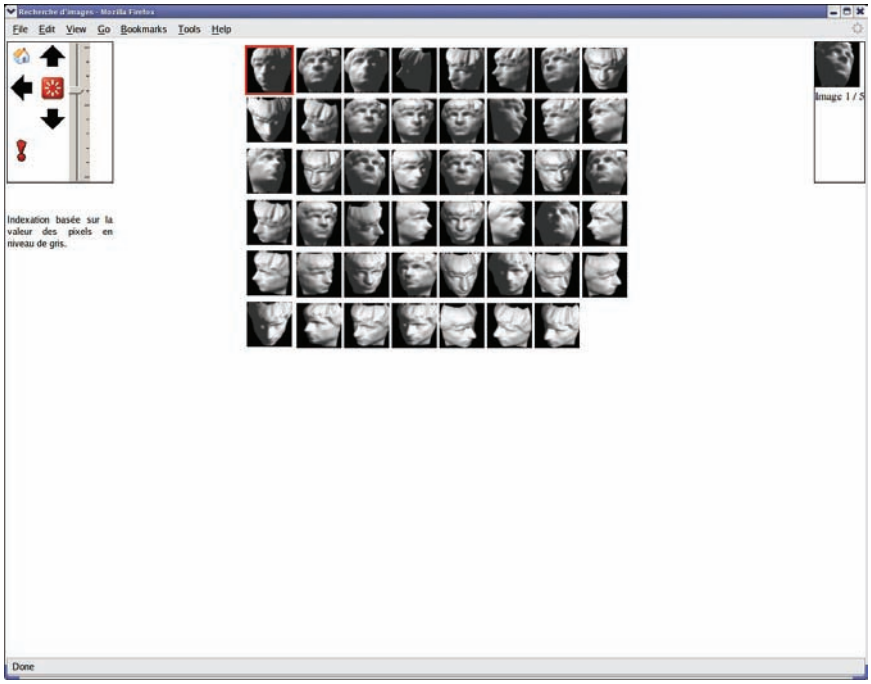


FIG. 5.4: Interface de navigation hiérarchique.

Le menu de navigation situé en haut et à gauche de l'écran permet de modifier le niveau du sommet courant, c'est à dire atteindre le sommet u^{i+1} ou u^{i-1}. La figure 5.5 présente l'utilisation de la flèche orientée vers le bas sur la position du sommet courant (cercle noir). Bien que le sommet courant change, l'image associée reste celle associée à l'élément $p^{(0)}$ de l'ensemble S. En revanche, le voisinage affiché dans le vue courante change car les sommets $r^{(1)}$ et $t^{(1)}$ apparaissent au niveau S_1 et sont reliés à $p^{(1)}$ par une arête.

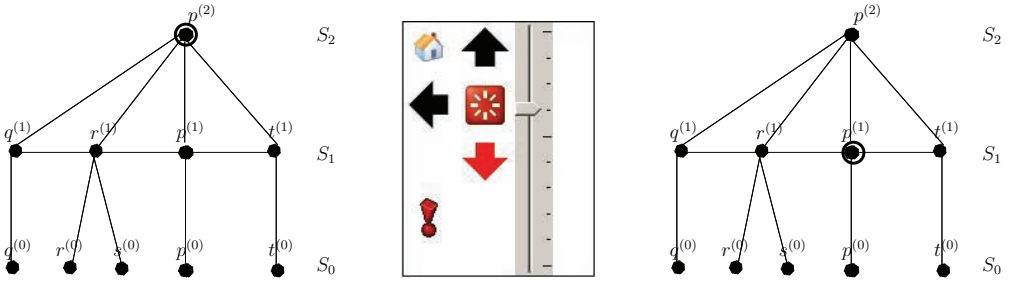


FIG. 5.5: Navigation depuis un élément situé au niveau 2 de la hiérarchie des centres discrets (sommet $p^{(2)}$) vers le même élément dans le niveau de hiérarchie inférieur (sommet $p^{(1)}$).

La figure 5.6 présente l'effet de la sélection du voisin $r^{(1)}$ du sommet courant $p^{(1)}$. Le sommet courant devient $r^{(1)}$, le voisinage change et contient les sommets $q^{(1)}$ et $p^{(1)}$ mais le niveau reste inchangé.

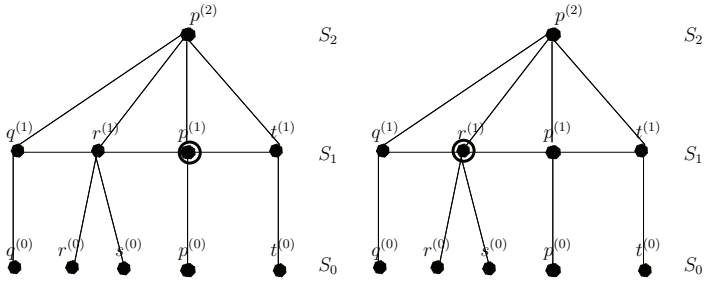


FIG. 5.6: Navigation depuis un élément vers son voisin au même niveau de la hiérarchie des centres discrets (niveau S_1).

La figure 5.7 présente l'effet de la flèche orientée vers le haut. Le sommet choisi comme parent pour $r^{(1)}$ devient le sommet courant. Le niveau passe de S_1 à S_2.

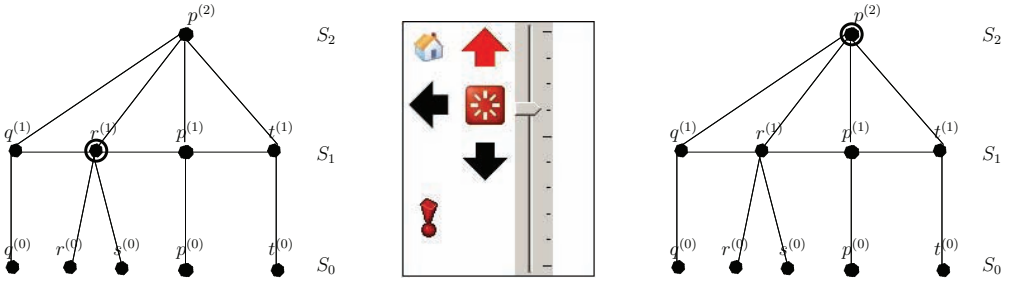


FIG. 5.7: Navigation depuis un élément situé au niveau 1 de la hiérarchie des centres discrets (sommet $r^{(2)}$) vers l'élément choisi pour le couvrir dans le niveau de hiérarchie supérieur (sommet $p^{(2)}$).

Le carré rouge permet de revenir à la racine de la hiérarchie. La flèche noire orientée vers la gauche permet de revenir à la position occupée précédemment. Le texte situé sous le menu de navigation indique le type d'indexation utilisé pour la collection d'images en cours.

5.3.2.1 Recherche exhaustive

On compare l'approche hiérarchique avec une interface basée sur la recherche linéaire dans une collection d'images. Cette méthode "naïve" mais facile a implémenter a été choisie car elle constitue un point de référence auquel on pourra comparer d'autres méthodes plus évoluées.

La figure 5.8 présente l'interface utilisateur présentant la totalité des images de la collection dans une vue unique. Les barres de défilement du navigateur permettent de parcourir la vue sur la collection et de sélectionner la cible recherchée avec la souris.

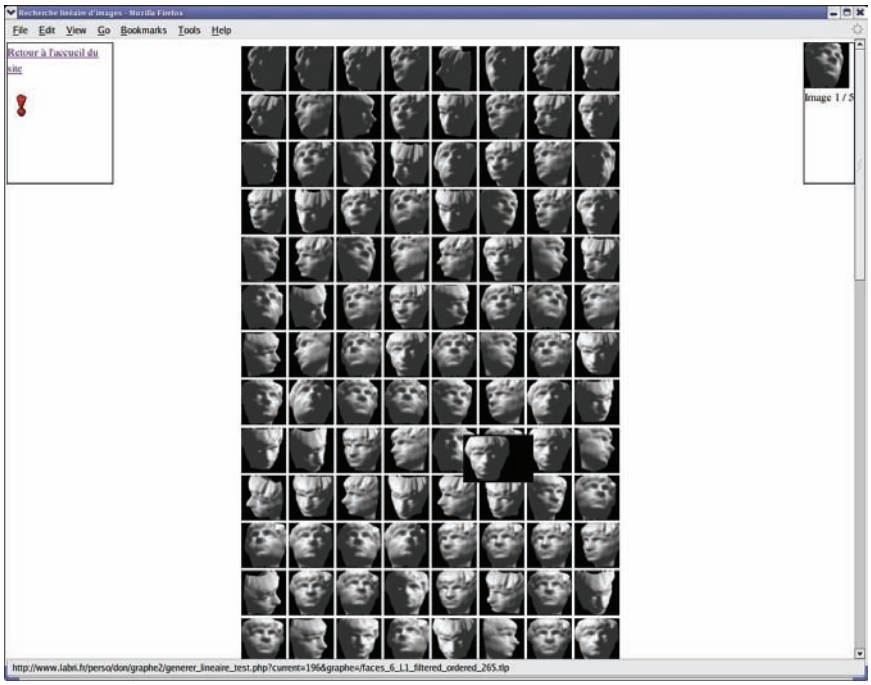


FIG. 5.8: Interface de recherche linéaire.

5.3.3 Descripteurs utilisés

La navigation dans l'interface utilisateur dépend de la faculté de l'utilisateur à choisir une image courante qui partage le plus de caractéristiques avec l'image recherchée parmi la liste d'images qui lui est présentée.

Ces caractéristiques étant exprimées sous la forme d'un descripteur, l'utilisateur doit être conscient de la composition du descripteur utilisé dans l'interface afin de ne pas se laisser guider par une caractéristique de l'image que n'utilise par le descripteur.

Pour comparer l'influence de "l'interprétabilité" du descripteur utilisé, nous avons employé différents descripteurs pour les différentes collections d'images utilisées.

5.3.3.1 Information de luminance

Pour la collection d'images contenant des rendus en niveaux de gris d'un modèle 3D de visage sous différentes pauses et position de la source lumineuse (cf. tableau 5.2, collection `faces_265`), le descripteur utilisé est l'ensemble des échantillons qui forment l'image.

Chaque image a une dimension de 64×64 pixels et l'intensité lumineuse de chaque pixel est exprimé sous la forme d'une valeur numérique dans l'intervalle $[0, 1]$. Chaque image peut donc être considérée comme un vecteur à 4096 composantes. La mesure de dissimilarité utilisée est la norme $L1$ dans l'espace des valeurs de ces vecteurs.

5.3.3.2 ISO/MPEG-7 CLD

Pour indexer les images naturelles, nous avons utilisé le descripteur MPEG7 Color Layout Descriptor (cf. définition du CLD dans la section 1.1.4.2) avec la totalité des coefficients fréquentiels (64 pour la luminance, 64 pour la chrominance rouge et 64 pour la chrominance bleu) a été utilisé. On peut comparer deux images sur la base de ce descripteur en considérant que des images similaires au sens du CLD, ont une disposition approximativement similaire des couleurs dans l'image. Cette correspondance approximative des régions de l'image s'explique par la réduction de résolution des images qui intervient lors de l'extraction du descripteur. La mesure de dissimilarité utilisée est la norme $L2$ dans l'espace des vecteurs formant les descripteurs CLD. Une pondération a été appliquée pour tenir compte de la sensibilité plus importante de la vision humaine aux basses fréquences (cf. section 1.1.4.2).

5.3.3.3 Descripteur ID

L'objectif de ce descripteur ID, pour "Interpretable Descriptor", est de proposer un descripteur qui soit plus interprétable que le descripteur CLD. Le descripteur ID est composé d'un vecteur numérique à six composantes appartenant à l'espace $[0, \infty[\times [0, 1] \times [0, \infty[\times [0, 1] \times [0, 1] \times [0, 1]$ correspondant aux caractéristiques suivantes :

- *NB_VISAGES* : nombre de visages présents dans l'image. Ce nombre a été déterminé par indexation manuelle de la collection d'images.
- *TAILLE_VISAGE* :Rapport de la surface de l'image occupée par un visage sur la surface totale de l'image.
- *NB_REGIONS* : nombre de régions dans l'image. Ce nombre a été déterminé après une segmentation couleur de l'image par quantification couleur.
- *LUMINANCE* : valeur de la luminance moyenne des couleurs dominantes. Les couleurs dominantes sont déterminées par la méthode d'extraction du descripteur MPEG7 Dominant Color Descriptor, présentée dans la section 1.1.4.2.
- *SATURATION* : valeur de la saturation maximum parmi les couleur dominantes de l'image. Cette valeur exprime la pureté de la couleur. Nous avons extrait cette valeur après conversion des couleurs dominantes de l'espace couleur RGB vers HSV. La saturation correspond à la composante S de cet espace couleur.
- *TEINTE* : valeur de la teinte associée à le couleur dominante la plus saturée. La teinte correspond à la composante V de l'espace couleur HSV et indique le type de couleur. Elle s'exprime sous la forme d'un angle, $v * 2\pi$ où l'angle 0 correspond à la teinte rouge, $\pi/2$ à la teinte verte, π à la teinte bleu, $3\pi/2$ à la teinte violette.

Afin de limiter les erreurs dans la détection de la teinte la plus saturée, l'interface utilisateur présente un bandeau horizontal, dont la couleur correspond à la saturation et la teinte stockée dans le descripteur, autour de la version agrandie de chaque image. La figure 5.9 présente un exemple d'image agrandie et du bandeau coloré avec la saturation et la teinte du descripteur associé.

La mesure de dissimilarité utilisée est la norme $L1$.

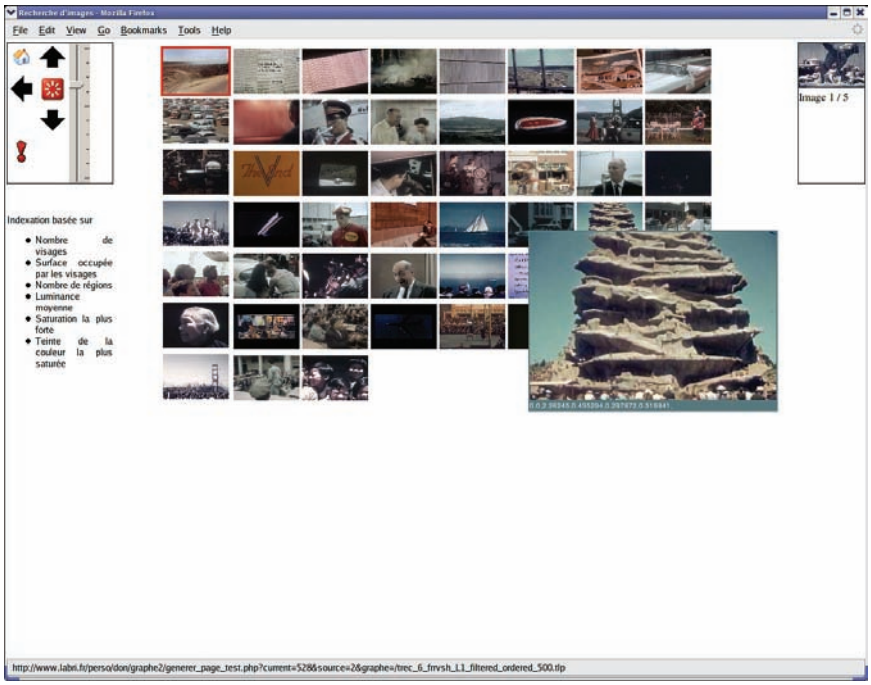


FIG. 5.9: Affichage de la teinte la plus saturée associée à chaque image.

Définition 65 (Mesure de dissimilarité entre descripteurs ID)
Soit S un ensemble de $|S| = n$ descripteurs ID. Soient $u = (u_1, u_2, u_3, u_4, u_5, u_6)$, $v = (v_1, v_2, v_3, v_4, v_5, v_6) \in S$ deux descripteurs ID dont les composantes correspondent aux indices visuels suivants :

- u_1, v_1 : NB_VISAGE,
- u_2, v_2 : TAILLE_VISAGE,
- u_3, v_3 : NB_REGIONS,
- u_4, v_4 : LUMINANCE,
- u_5, v_5 : SATURATION,
- u_6, v_6 : TEINTE.

La mesure de dissimilarité entre u et v, notée $\delta(u, v)$ est définie par :

$$\begin{aligned}\delta(u,v) &= \left|\frac{\log(1+u_1)-\log(1+v_1)}{\log(1+V_{max})}\right| + |u_2 - v_2| + \left|\frac{\log(1+u_3)-\log(1+v_3)}{\log(1+R_{max})}\right| \\ &\quad + |u_4 - v_4| + |u_5 - v_5| + |u_6 - v_6| \end{aligned} \tag{5.6}$$

V_{max}, resp. R_{max}, désigne le nombre de visages, resp. le nombre de régions, maximum dans les descripteurs de l'ensemble S. Soient $u = (u_1, u_2, u_3, u_4, u_5, u_6)^T$, $v = (v_1, v_2, v_3, v_4, v_5, v_6)^T \in S$ deux descripteurs ID

Comme on peut le constater dans la définition 65, les valeurs du descripteur qui correspondent à des cardinaux (nombre de visages et nombre de régions) sont modifiées par la fonction $f(x) = \log(1 + x)$. Cette modification de la valeur associée à ces indices visuels est utilisée afin de tenir compte des caractéristiques de la perception humaine et plus précisément de la non-linéaire de la perception du nombre d'objets. Cette caractéristique de la perception peut être évaluée en estimant la fonction psychophysique associée à la perception du nombre d'objets dans une image.

5.3.3.4 Fonction psychophysique de la perception du nombre d'éléments

On demande à l'utilisateur d'effectuer des déplacements dans un graphe dont la structure dépend de la mesure de dissimilarité calculée d'après le contenu des images. Cette mesure de dissimilarité est calculée à partir des indices visuels extraits des images. Ces valeurs correspondent à des caractéristiques mesurables, comme le nombre de régions ou la luminance moyenne de l'image. Idéalement, on souhaite que la mesure de dissimilarité reflète la perception de l'utilisateur pour lui permettre d'interpréter au mieux les caractéristiques composant le descripteur. Nous souhaitons que la mesure de dissimilarité relie de manière linéaire et uniforme, la valeur du descripteur à la perception de l'utilisateur.

Soient deux paires d'images, (a, b) et (c, d), dont on a extrait le nombre de régions. (a, b) et (c, d) ont le même espacement sur l'échelle des valeurs du descripteur, (i.e. $\delta(a, b) = \delta(c, d)$) et sont situées en des points éloignés de l'espace des valeurs (par exemple, a contient 20 régions et c contient 200 régions). Si l'espace des valeurs n'est pas perceptiblement uniforme, alors la différence perçue entre (a, b) et (c, d) ne sera pas la même bien que la mesure de dissimilarité associée soit identique. Pour mesurer les caractéristiques de la perception humaine du nombre de régions, ou du nombre d'objets dans une image, on utilise la technique de "Maximum Likelihood Difference Scaling" (MLDS) décrite dans l'Annexe A.

La psychophysique est la discipline qui étudie *quantitativement* les relations entre stimulations et sensations [24]. La *fonction psychophysique* associée à un stimulus est une fonction mathématique qui relie l'intensité physique du stimulus à la sensation perçue. Dans notre exemple le stimulus est constitué par une image contenant des objets et l'intensité du stimulus est le nombre d'objets dans l'image. On souhaite donc déterminer les caractéristiques de cette fonction.

Perception du nombre d'éléments L'expérience réalisée consiste à évaluer la fonction psychophysique associée à la perception du nombre d'éléments dans une image, notée $\Psi(x)$, qui associe au stimulus d'intensité x une valeur psychophysique dans l'intervalle $[0, 1]$.

Nous avons généré $S = 7$, stimuli composés d'images contenant respectivement 8, 16, 32, 64, 128, 256 et 512 glyphes représentant le caractère '$*$' en blanc sur fond noir. Chaque glyphe utilise une surface de 8×8 pixels sans chevauchement. La figure 5.10 présente les 7 stimuli utilisés pour cette expérience. Chaque image a une dimension de 360×360 pixels.

La totalité des $\binom{S}{4}$ quadruplets de stimuli ordonnés sont présentés séquentiellement à l'observateur sur un écran, la première paire étant affichée dans la moitié supérieure de l'écran et la deuxième paire dans la moitié inférieure. L'ordre des images dans chaque paire ainsi que la position des paires sur l'écran sont déterminés de manière aléatoire uniforme.

Au cours de l'expérience, l'observateur indique si la paire du haut ($R = 0$) ou celle du bas ($R = 1$) lui semble la plus différente. L'ensemble, $(R_i)_{1 \leq i \leq \binom{S}{4}}$, des réponses fournies par l'observateur est ensuite utilisé par la procédure qui estime la valeur de la fonction Ψ aux S points correspondant aux stimuli.

La figure 5.11 présente l'estimation de la fonction psychophysique d'un observateur ayant réalisé l'expérience.

La pente de la courbe de la figure 5.11 indique la sensibilité de la perception dans l'intervalle de valeur correspondant. Une pente supérieure à 1 indique qu'une faible différence entre les valeurs de deux stimuli conduira à une différence plus importante sur l'échelle perceptuelle. Cela signifie que dans la zone de valeurs correspondant à une forte pente de la courbe, l'acuité est forte alors que dans la zone où la pente est faible, l'acuité plus faible.

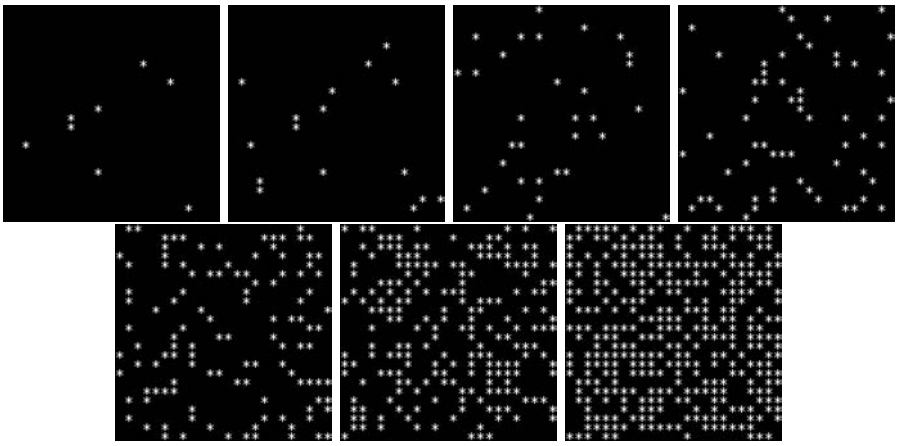


FIG. 5.10: Les sept stimuli utilisés pour l'estimation de la fonction psychophysique de la perception du nombre d'éléments dans une image.

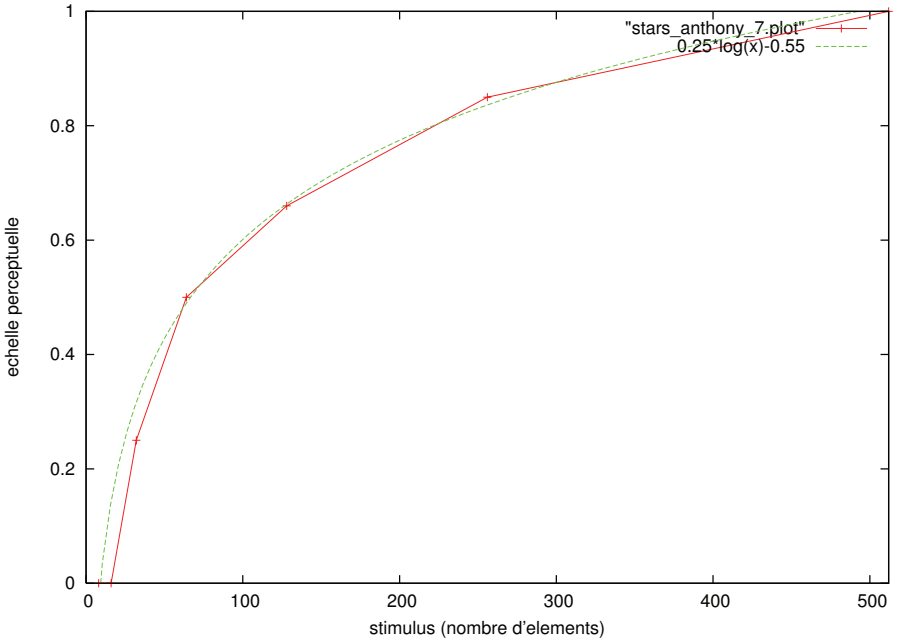


FIG. 5.11: Estimation de la fonction psychophysique d'un observateur pour la perception du nombre d'éléments dans l'image.

La figure 5.11 indique clairement que la perception du nombre d'éléments dans l'image n'est pas linéairement proportionnelle au nombre d'éléments dans l'image : L'échelle des valeurs n'est pas perceptiblement uniforme. La perception du nombre d'éléments est plus précise dans le domaine des petites cardinalités (entre 0 et 50 éléments) que pour les valeurs plus importantes. L'allure de la courbe suggère une fonction psychophysique logarithmique pour la perception du nombre d'éléments. La figure 5.11 montre le tracé d'une fonction d'équation $f(x) = a \log_b(x) + c$ approximant la fonction psychophysique ($a = 0,25$; $b = e$ et $c = 0,55$).

Nous utilisons ce résultat dans la mise au point du descripteur ID. La première ainsi

que la troisième composante de ce descripteur représentent le cardinal de certains attributs de l'image : le nombre de visages ainsi que le nombre de régions. Nous supposons que la perception de ces éléments particuliers de l'image utilise une fonction psychophysique (cf. annexe A) *similaire* à celle de la figure 5.11. C'est pourquoi nous appliquons la fonction $f(x) = \log(1 + x)$ aux valeurs originales de la première et de la troisième composante de chaque descripteur ID ($\log(1 + x)$ a été utilisé pour garantir que $f(x)$ soit définie en 0).

Imaginons deux images au contenu texturé, contenant respectivement 250 et 350 régions. La prise en compte de la fonction psychophysique permet de réduire la contribution injustifiée de la troisième dimension du descripteur à la mesure de dissimilarité entre les deux images. En effet, il semble naturel de considérer de telles images comme des "images texturées" et donc de les considérer comme similaires du point de vue du nombre de régions.

5.3.4 Données utilisées

Nous utilisons deux collections d'images différentes :

- Images-clé extraites du corpus de documents vidéo de la campagne d'évaluation TREC Video, de l'année 2002 [123]. Cette collection correspond au préfixe `trec` dans le Tableau 5.2.
- Différents rendus d'un modèle 3D de visage selon différentes conditions de pose et d'éclairage. Ce jeu de données a été produit par les auteurs de l'algorithme Isomap [152]. Cette collection correspond au préfixe `face` dans le Tableau 5.2. Nous avons sélectionné 265 images parmi les 698 que compte le jeu de données original. Les images indistinguables ont été supprimées de ce jeu de données.

Le Tableau 5.2 présente les caractéristiques des collections utilisées pour les expériences :

- *Taille* désigne le nombre d'images dans la collection traitée, ce paramètre influence le temps requis pour accomplir chaque requête.
- *Dimensions* correspond au nombre de dimensions requis pour représenter les caractéristiques extraites des images, c'est une borne supérieure de la dimension intrinsèque des données.
- *Descripteur* indique le type de descripteur utilisé. Ce critère intervient dans le caractère "interprétable" de l'indexation utilisée.
- *Norme* désigne le type de mesure de dissimilarité qui est utilisé.
- *Voisinage* indique la valeur du paramètre c utilisé pour définir le voisinage d'un sommet avec les sommets du même niveau.

Le choix des différents jeux de données est motivé par les caractéristique décrites ci-dessous :

- `trec_CLD_500` : grande dimension, interprétation difficile du descripteur.
- `faces_265` : faible dimension intrinsèque des données, descripteur très bas niveau.
- `trec_ID_500` : faible dimension intrinsèque des données et interprétation facile du descripteur.
- `trec_ID_1000` : faible dimension intrinsèque des données et interprétation facile du descripteur. Augmentation du nombre d'éléments.

Nom	Taille	Dim.	Contenus	Descripteur(s)	Norme
`trec_CLD_500`	500	12	Images-clé extraites de documentaires TREC 2002	CLD	L2 pondérée
`faces_265`	265	4096	Rendus en niveaux de gris d'un modèle 3D de visage sous différentes pauses et position de la source lumineuse	Luminance	L2
`trec_ID_500`	500	6	Images-clé extraites de documentaires TREC 2002	Descripteur ID	L1
`trec_ID_1000`	1000	6	Images-clé extraites de documentaires TREC 2002	Descripteur ID	L1

Tab. 5.2: Jeux de données utilisés pour les expériences de navigation.

5.3.5 Méthodologie

5.3.5.1 Déroulement des tests

Après une explication des différents types d'indexation et des types d'interfaces, les utilisateurs-test ont été invités à manipuler librement chacune des interfaces sur chacune des collections. A l'issue de cette étape d'entraînement, l'utilisateur crée un compte l'identifiant de façon unique puis commence à réaliser les recherches dans l'ordre indiqué dans le Tableau 5.3.

Chaque expérience correspond à l'enchaînement d'une série de cinq recherches avec l'interface indiquée. Chaque recherche est limitée à un temps maximum, t_{max}, fixé à 120 secondes. Après cette limite, le système de test passe automatiquement à la recherche suivante. L'utilisateur a également la possibilité d'abandonner la recherche en cours et de passer à la suivante.

Dans le cadre de l'interface linéaire, le succès d'une recherche correspond à la sélection de l'image recherchée par l'utilisateur. Dans le cadre de l'interface hiérarchique, le succès d'une recherche correspond au fait que l'image recherchée devient l'image courante.

A l'issue de la série de recherches, l'utilisateur-test est invité à donner une note entre zéro et dix mesurant le niveau d'efficacité perçu de la tâche réalisée. Un commentaire optionnel peut également être saisi.

L'analyse des résultats des expériences réalisées sur les différents jeux de données vont nous permettre de conclure sur les points suivants :

– L'interface hiérarchique est-elle plus efficace que l'interface linéaire ? Nous répondrons à cette question en comparant les résultats de la paire d'expériences

	Collection	Descripteur	Interface
Expérience 1	`faces_265_lineaire`	Luminance	Recherche exhaustive.
Expérience 2	`faces_265_2_lineaire`	Luminance	Recherche hiérarchique. $c = 2$.
Expérience 3	`faces_265_6_hierarchique`	Luminance	Recherche hiérarchique.$c = 6$.
Expérience 4	`trec_ID_500_lineaire`	ID	Recherche exhaustive.
Expérience 5	`trec_CLD_500_hierarchique`	CLD	Recherche hiérarchique. $c = 6$.
Expérience 6	`trec_ID_500_hierarchique`	ID	Recherche hiérarchique. $c = 6$.
Expérience 7	`trec_ID_1000_hierarchique`	ID	Recherche hiérarchique. $c = 6$.

Tab. 5.3: Expériences réalisées.

`faces_265_6_lineaire` et `faces_256_6_hierarchique` ainsi que ceux de `trec_ID_500_6_lineaire` et `trec_ID_500_6_hierarchique`.

– Dans le cadre de l'interface hiérarchique, l'indexation ID est-elle plus efficace que l'indexation basée sur CLD ? Nous répondrons à cette question en comparant les résultats de la paire d'expériences `trec_ID_500_6_hierarchique` et `trec_CLD_500_6_hierarchique`.
– Dans le cadre de l'interface hiérarchique, le paramètre de voisinage théorique rend t'il la recherche plus efficace qu'un paramètre de voisinage plus faible ? Nous répondrons à cette question en comparant les résultats de la paire d'expériences `faces_256_6_hierarchique` et `faces_256_2_hierarchique`.

5.3.5.2 Expériences

Le Tableau 5.3 décrit les expériences réalisées. Chacune des sept expériences consiste à chercher une série de cinq images dans la collection associée avec l'interface proposée.

5.3.5.3 Cibles

Les cibles sélectionnées pour réaliser les recherches ont été tirées de manière aléatoire uniforme dans la collection associée. Ces cibles sont les mêmes pour tous les utilisateurs. Les cibles sont présentées dans l'ordre des figures 5.12, 5.13 et 5.14.

Fig. 5.12: Cibles pour les expériences 1,2 et 3.

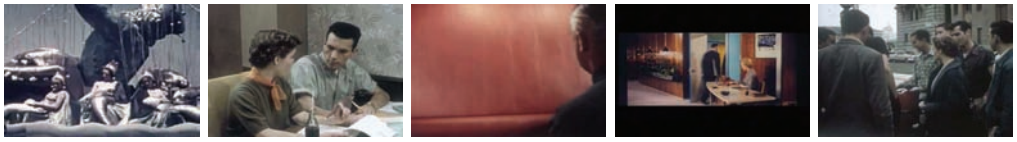


FIG. 5.13: Cibles pour les expériences 4,5 et 6.

FIG. 5.14: Cibles pour l'expérience 7.

5.3.5.4 Données recueillies

Pour chaque recherche utilisant l'interface linéaire, nous avons enregistré l'issue de la recherche (cible trouvée, abandon ou temps écoulé), le temps total depuis le début de la recherche (en secondes).

Pour chaque recherche avec l'interface hiérarchique, nous avons recueilli l'issue de la recherche (cible trouvée, abandon ou temps écoulé), le temps total depuis le début de la recherche (en secondes) et le chemin emprunté par l'utilisateur-test dans le graphe.

Les commentaires des utilisateurs ainsi que les notes qu'ils ont attribué à chaque interface ont également été enregistrés.

5.3.6 Choix d'implémentation

Dans cette sous-section, nous présentons les choix techniques retenus pour la construction de la hiérarchie des centres discrets ainsi que pour la réduction du nombre de voisins dans le NAV-GRAPHE.

5.3.6.1 Construction de la hiérarchie des centres discrets

Nous utilisons l'algorithme Cover Tree [21] pour construire l'arbre d'échantillonage T. Ce choix est motivé par l'efficacité de la construction.

Lemme 5.4 ([21])
Soit (S, δ) un espace métrique, soit $n = |S|$ et soit c_d la constante doublante de (S, δ). La complexité en temps de la construction d'un Cover Tree est en $O(c_g^6 n \log n)$.

L'expansion, notée c_g, caractérise la dimension intrinsèque des données traitées. Cette valeur conditionne le nombre de points qui peuvent être couverts par un parent. Il existe une relation entre la constante doublante c_{dd} et l'expansion c_g. Si l'expansion est bornée, alors la constante doublante l'est aussi [92]. L'inverse n'est cependant pas vrai.

Dans [21], les auteurs prouvent que ce nombre est borné par c_g^4. Cette caractéristique est très importante dans le cadre de la navigation hiérarchique car le nombre d'éléments constituant le voisinage d'un sommet au niveau i est proportionnel à la taille de l'échantillon de niveau i. Plus le nombre de sommets couverts est important, plus la taille de l'échantillon de niveau inférieur augmente rapidement. Ainsi, le choix du nouveau sommet courant à chaque étape requiert d'examiner un nombre important de sommets voisins quand la dimension intrinsèque des données est grande.

5.3.6.2 Filtrage des voisins

On propose de réduire la taille de l'ensemble $N_i(u^{(i)})$ en supprimant les sommets du voisinage de chaque sommet $u^{(i)} \in S_i$ si ces sommets sont la racine d'un sous-arbre dont aucune feuille n'est couverte par u à distance 2^{i+1}.

Parmi l'ensemble $N_i(u^{(i)})$ des voisins d'un sommet $u^{(i)}$ au niveau $i1$, certains n'ont aucune feuille dans leur sous-arbre qui soit située dans la zone pouvant avoir $u^{(i+1)}$ comme ancêtre de niveau $i+1$. Dans la mesure où l'examen des voisins de $u^{(i)}$ concerne la recherche d'un élément proche de $u^{(i)}$, les voisins dont le sous-arbre ne contient aucune feuille située à distance inférieure à 2^{i+1} de $u^{(i)}$ sont supprimés de $N_i(u^{(i)})$. Ceci permet de réduire d'avantage la taille du voisinage et le nombre d'éléments à examiner pour le choix du prochain élément courant.

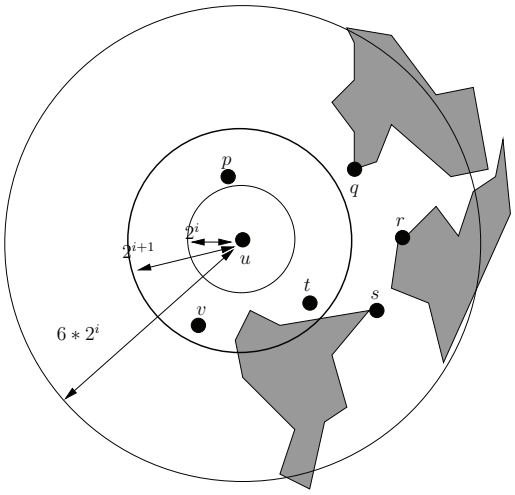


FIG. 5.15: Filtrage du voisinage de u au niveau i.

La figure 5.15 illustre le filtrage du voisinage du sommet $u^{(i)}$. Les feuilles des sous-arbres enracinés aux sommets $q^{(i)}$ et $r^{(i)}$ sont toutes à distance supérieure à 2^{i+1} de u. Les sommets $q^{(i)}$ et $r^{(i)}$ sont donc supprimés de l'ensemble $N_i(u^{(i)})$. Les points de l'espace métrique associés aux sommets $p^{(i)}, v^{(i)}$ et $t^{(i)}$ sont à distance inférieure à 2^{i+1} de $u^{(i)}$. Comme ces points seront présents au niveau le plus bas, celui des feuilles, ils ne peuvent donc pas être supprimés. Enfin, le sous-arbre enraciné en $s^{(i)}$ possède des feuilles à distance inférieure à 2^{i+1} de $u^{(i)}$, $s^{(i)}$ est donc conservé dans $N_i(u^{(i)})$.

5.3.6.3 Interface Web

Les expériences ont été réalisées via une interface Web programmée en PHP. Les graphes de navigation ainsi que les informations issues des expériences sont stockées sous forme de tables de base de données MySQL.

Le temps total utilisé par le serveur pour générer chaque vue est enregistré avec les données de la recherche en cours. Cela permet de tenir compte des temps de chargement des pages à chaque action sur l'interface. Ces valeurs sont soustraites du temps de recherche total pour chaque requête

5.4 Résultats

La série de sept expériences a été complétée par onze utilisateurs-test (une femme et dix hommes). Les onze personnes sont des utilisateurs réguliers de l'outil informatique.

Chaque expérience a été conduite en notre présence afin de procéder à une présentation des deux interfaces et de répondre aux questions des utilisateurs-test. Un travail supplémentaire sur la présentation des interfaces et des types d'indexation permettrait de mener cette expérience sur un plus grand nombre d'individus via le Web.

Nous dégageons les tendances données par les résultats des expériences réalisées. Nous concluons notamment sur l'influence du paramétrage du voisinage, il semble qu'un effet de seuil intervienne et minimise l'impact du paramètre de voisinage sur les performances de navigation. Enfin, nous discutons de l'influence du type d'indexation sur les performances de recherche.

5.4.1 Efficacité mesurée

Cette sous-section présente les données recueillies lors des expériences afin de comparer l'efficacité des recherches effectuées dans les sept conditions expérimentales. Nous exprimons l'efficacité par ·

- le taux de réussite des recherches effectuées avec chaque couple (interface, descripteur),
- le temps médian par requête en secondes,
- le temps médian par requête réussie en secondes.,

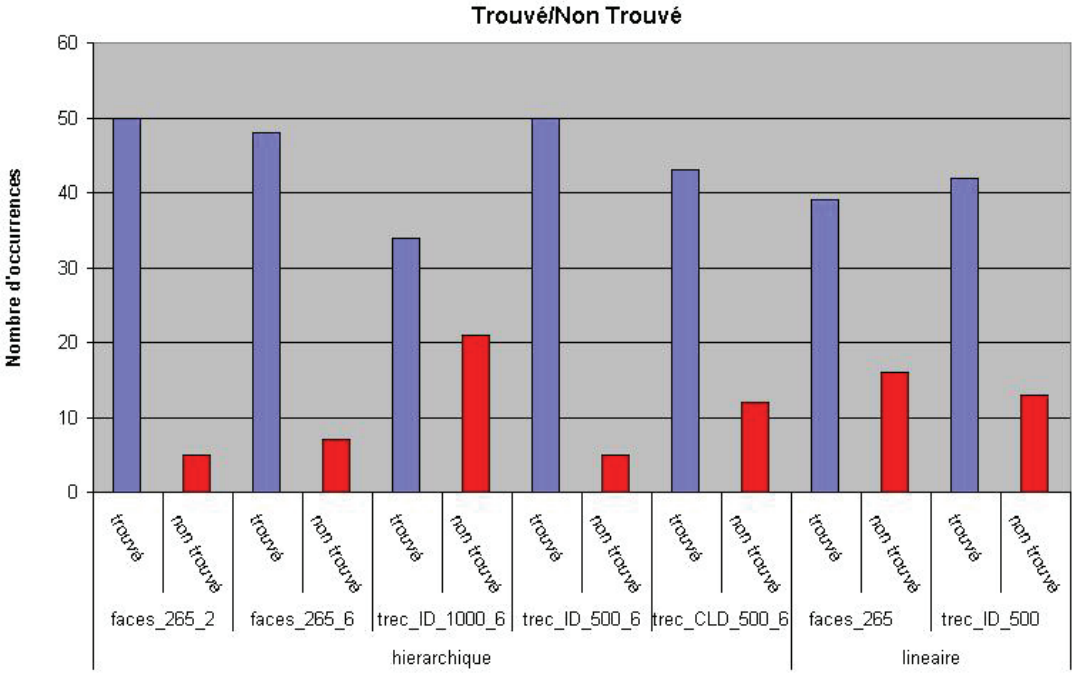
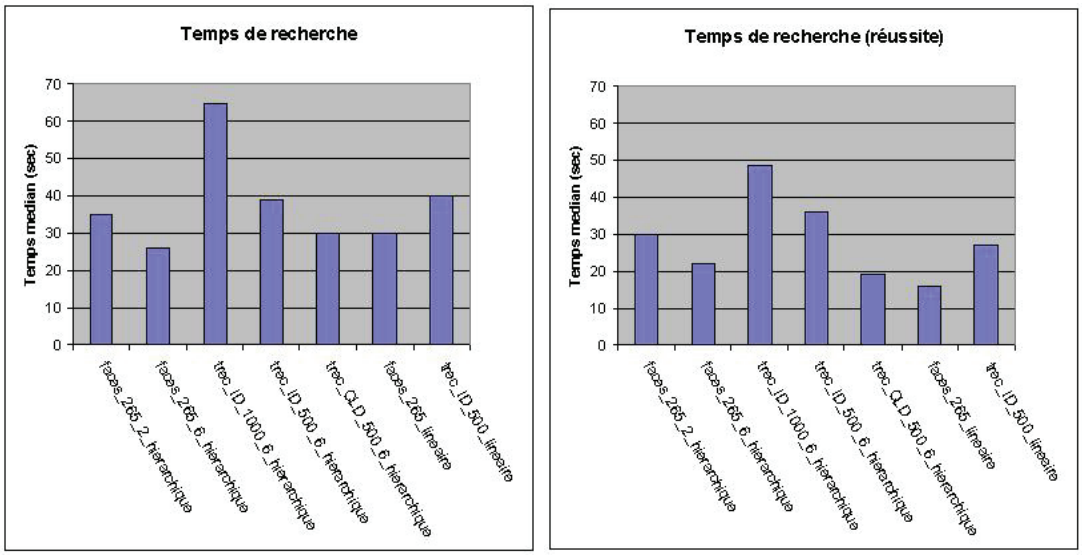


FIG. 5.16: Mesure d'efficacité (réussite) des recherches en fonction de la collection d'images et de l'interface utilisée.

La figure 5.16 détaille l'effectif des recherches ayant aboutie à la cible recherchée (étiquette `trouvé` en bleu) ainsi que l'effectif des recherches ayant échoué (étiquette `nontrouvé`

en rouge). Pour la collection `faces`, la réussite des interfaces hiérarchiques (`faces_265_2_hierarchique` et `faces_265_6_hierarchique`) est supérieure à celle de l'interface linéaire (`faces_265_6_lineaire`). Pour la collection `trec`, l'interface lineaire et l'interface hierarchique utilisant l'indexation par le CLD (`trec_ID_500_6_lineaire` et `trec_CLD_500_6_hierarchique`) conduisent à des résultats similaires alors que l'interface hiérarchique avec l'indexation ID (`trec_ID_500_6_hierarchique`) conduit à une réussite supérieure.



a) b)

FIG. 5.17: Mesure d'efficacité (temps) des recherches en fonction de la collection, du descripteur et de l'interface utilisée. a) Temps médian par recherche. b) Temps médian par recherche réussie.

La figure 5.17.a) présente le temps médian par recherche. Ces temps étant influencés par les recherches infructueuses, conduisant à un temps maximum de 120 secondes, nous présentons également, dans la figure 5.17.b), le temps médian de recherche pour les recherches menées à terme.

On peut constater (figure 5.17.a)) que pour la collection `faces`, on constate un temps de recherche médian inférieur pour le paramétrage du voisinage utilisant la valeur $c = 6$ par rapport à la version utilisant $c = 2$.

5.4.1.1 Taille de la collection

L'expérience utilisant l'interface hiérarchique avec 1000 images indexées avec le descripteur ID , présente un plus grand nombre d'échecs que celle utilisant la collection de 500 images. Nous analyserons les raisons de ce résultat dans la section 5.5. Nous soulignons cependant qu'en cas de réussite (cf. figure 5.17.b)), l'augmentation du temps médian pour compléter la recherche passe de 38 secondes à 48 secondes, soit moins qu'un facteur 2. Cette information sera également discutée dans la section 5.5.

5.4.1.2 Efficacité de l'interface hiérarchique

Les temps médian de recherche (figure 5.17.a)) montrent que pour les tailles des jeux de données utilisés (265 images pour la collection `faces` et 500 images pour la collection `trec`), les temps de recherche sont comparables.

Nous concluons donc que d'après les résultats de l'expérience, l'interface hiérarchique présente une efficacité similaire à l'approche linéaire selon le critère du temps mais que le succès dans l'issue des recherches est plus important avec l'interface hiérarchique.

5.4.1.3 Utilisation du descripteur CLD vs descripteur ID

La figure 5.16 indique un net avantage pour l'indexation ID par rapport à l'indexation par CLD, le nombre d'échecs étant moins important avec `trec_ID_500_6` qu'avec `trec_CLD_500_6`. Ce résultat plaide en faveur de l'indexation ID de la collection d'images puisque les autres paramètres sont identiques (type et taille de la collection d'images, cibles utilisées, type d'interface et paramétrage de l'interface).

L'indexation ID permet une meilleure interprétation des caractéristiques extraites de l'image par le processus d'indexation. Par opposition, il est plus difficile d'interpréter le descripteur Color Layout Descriptor dans la mesure où il représente des composantes fréquentielles associées à chaque composante couleur de l'image.

Nous pensons que l'interprétabilité du descripteur est une partie essentielle de la réponse à apporter pour permettre la navigation dans ce type d'interface. Le fossé sémantique ("semantic gap") est un des obstacles les plus importants dans le domaine des systèmes CBIR (Content-Based Indexing and Retrieval). Ce problème est central dans notre approche. En effet, l'image partageant le plus de caractéristiques avec la cible recherchée doit pouvoir être désignée le plus facilement possible en examinant le contenu des images. L'indexation proposée ici semble donc compatible avec cet objectif.

5.4.1.4 Impact de la dimension intrinsèque des données

L'interprétabilité du descripteur n'est pas la seul critère conditionnant le succès de la navigation hiérarchique. Les recherches sur la base de visages 3D sont basées sur l'information stockée dans l'image elle-même, une information difficilement interprétable. Malgré ce handicap a priori, la réussite des recherches hiérarchiques utilisant cette indexation est toutefois supérieure celles de la recherche linéaire (cf. figure 5.16, `faces_265_6_lineaire` et `faces_265_6_hierarchique`).

Nous expliquons cela par la nature très particulière de la collection d'images utilisées. Les auteurs de ce jeu de données indiquent dans [152] que l'ensemble des images utilisée forme une surface dans l'espace des valeurs $[0,1]^{4096}$. Les éléments de la collection étant uniformément répartis sur cette surface et la distance entre deux points voisins étant faible, il est possible d'atteindre n'importe quel point de cette surface depuis n'importe quelle autre position. De plus, la faible dimension intrinsèque des données (deux degrés de liberté pour la pose du visage et un degré de liberté pour la position de l'éclairage) permet d'afficher autour de tout point courant, au moins un voisin par "direction" intrinsèque (i.e. un voisin dont la pause est plus ou moins tournée vers la gauche, plus ou moins tournée vers le haut et avec la lumière positionnée plus ou moins à gauche).

La répartition homogène des points dans l'espace de description est donc un autre critère important pour permettre une bonne navigabilité dans le jeu de données. De manière symétrique, imaginons qu'une zone de l'espace de description comporte une plus grande concentration de points que les autres. Alors si les images associées à cette zone ont un

contenu sémantique différent les unes des autres, cela signifie que le descripteur utilisé n'est pas suffisamment discriminant et on pourrait qualifier ces images de "difficiles" à trouver.

5.4.2 Déplacements dans les interfaces hiérarchiques

Cette sous-section présente les caractéristiques des déplacements dans le NAV-GRAPHE effectués par les utilisateurs avec les interfaces hiérarchiques. Ces mesures permettent de mesurer l'impact du coefficient de voisinage, du type d'indexation et du type de collection sur la navigation.

La figure 5.18 présente la répartition des déplacements des utilisateurs au sein de chaque interface hiérarchique pour la totalité des recherches effectuées. Pour chaque interface, le nombre de clics moyen est donné. Le nombre moyen de remontées dans la hiérarchie (étiquette `haut`) donne une indication sur le nombre de cas où le sous-arbre en cours d'exploration est considéré comme insatisfaisant et où l'utilisateur juge nécessaire de généraliser la vue courante pour changer de sommet courant. Le nombre moyen de déplacements vers les niveaux inférieurs de la hiérarchie (étiquette `bas`) ainsi que le nombre de déplacement au sein du même niveau (étiquette `égal`) sont également donnés.

Dans le cas idéal d'une indexation parfaitement interprétable et d'un utilisateur ne commettant par d'erreurs, on observerait un nombre de déplacements vers le bas supérieur ou égal au nombre de déplacement par niveau. Le nombre de clics vers le haut serait nul.

Nous soulignons la valeur plus faible du nombre de remontées dans la hiérarchie avec le paramétrage du voisinage $c = 6$ (`faces_265_6`) qu'avec le paramétrage $c = 2$ (`faces_265_2`). On remarque également la valeur élevée de ce nombre pour la collection de mille images-clé (`trec_ID_1000_6`).

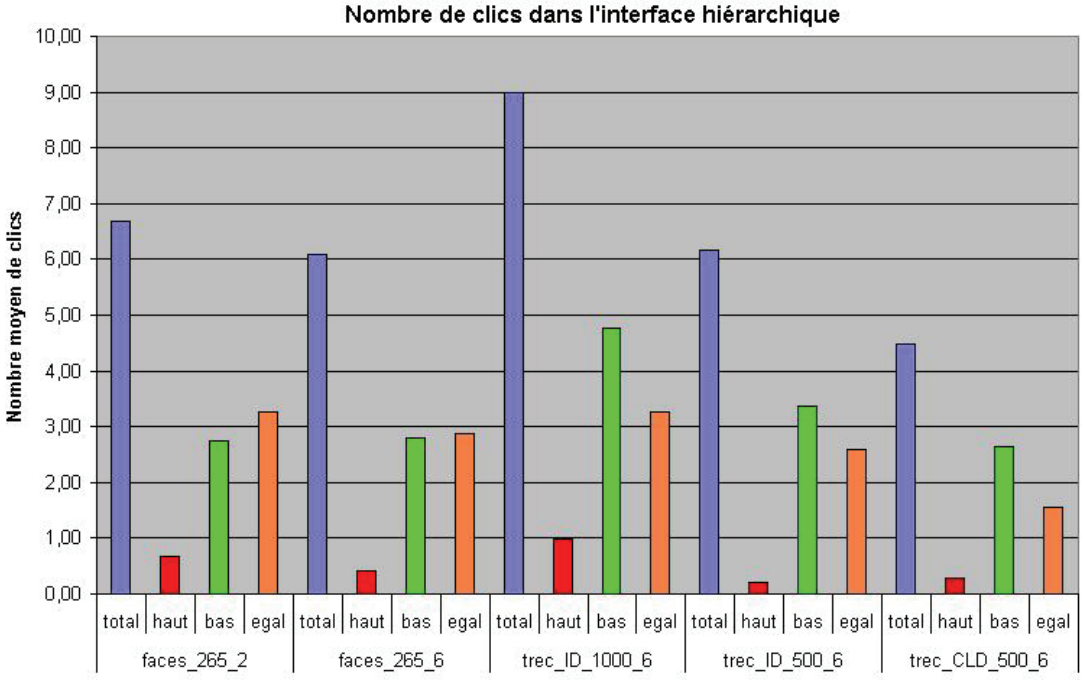


FIG. 5.18: Répartition des déplacements dans l'interface hiérarchique.

La figure 5.19 détaille les caractéristiques des déplacements effectués dans le voisinage du sommet courant. Pour chaque type de collection et d'indexation, on présente la moyenne et la médiane de la taille du voisinage (étiquette `size`), du temps, en secondes, entre deux clics consécutifs (étiquette `time`) et du rang de l'image sélectionnée dans la liste des voisins du sommet courant (étiquette `rank`). Ces informations sont commentées dans la section suivante.

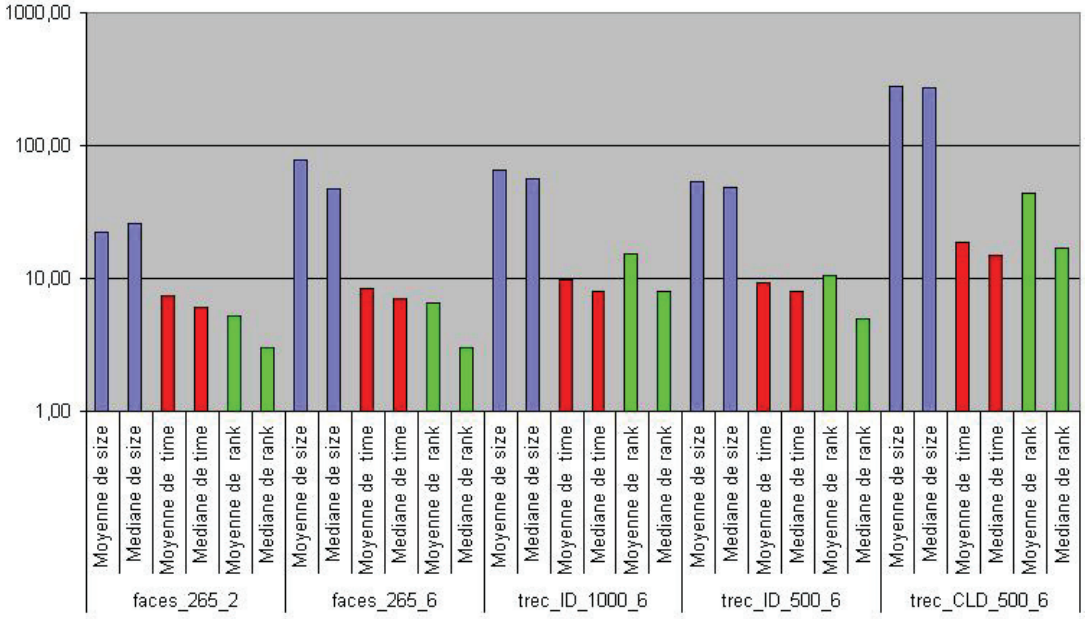


FIG. 5.19: Évolution des caractéristiques de déplacement dans le voisinage en fonction de la taille (étiquettes `size`) du voisinage et des caractéristiques de la collection.

5.4.2.1 Influence du voisinage

Taille du voisinage La figure 5.18 indique que le paramétrage théorique que nous avons présenté dans la section 5.4.2 induit un nombre de remontés dans la hiérarchie inférieur à celui obtenu avec un paramétrage plus restreint du voisinage. Cependant, la différence, entre les deux valeurs est faible et nous pouvons nous interroger sur la nécessité de conserver un voisinage si important autour du sommet courant ($c = 6$).

Corrélation avec le temps de décision De plus, la figure 5.19 indique que le temps de décision (temps par clic) et la position du voisin choisi dans la liste des voisins n'augmente pas aussi rapidement que la taille du voisinage proposé. On rappelle que la taille du voisinage dépend du paramétrage de c utilisé, et également du nombre d'éléments dans le niveau courant. Celui-ci dépend de la dimension intrinsèque des données. Dans le cas de la collection `trec_CLD_500_6`, bien que la taille du voisinage augmente de manière significative par rapport aux autres collections d'images, le temps de décision et le rang du voisin choisi augmentent peu.

Existence d'un seuil Nous pensons qu'il existe un effet de seuil, et que l'utilisateur ne consulte qu'un nombre restreint de voisins, indépendamment de la taille du voisinage. La valeur moyenne maximum rencontrée pour le rang auquel un voisin est choisi étant inférieure à cinquante, nous proposons de limiter la taille du voisinage à un nombre constant de voisins sans nécessairement dégrader les performances du processus de recherche. Cela permettrait ainsi de limiter la taille de l'affichage ainsi que le stockage de la structure de données.

5.4.3 Efficacité perçue

Nous présentons les notes attribuées par les utilisateurs à l'issue de chaque série de cinq recherches pour chacune des sept expériences proposées. Cette note, située entre zéro et dix, mesure l'efficacité de l'utilisateur dans la réalisation des recherches demandées. Celle-ci permet de vérifier si la perception de l'utilisateur correspond à ses performances quantitatives.

La figure 5.20 présente la moyenne et l'écart-type des notes attribuées par les onze utilisateurs-test. On remarque que les recherches dans la collection `faces` paraissent plus efficaces avec les interfaces hiérarchiques qu'avec l'interface linéaires, bien que pour cette dernière, l'étalement des notes soit plus important. La même tendance peut être observée pour la collection `trec` avec l'indexation ID. De plus, l'efficacité perçue varie peu pour la collection de taille 1000 par rapport à la collection de taille 500. En revanche, la navigation dans la collection indexée par le descripteur CLD est jugée moins efficace que l'indexation utilisant le descripteur ID.

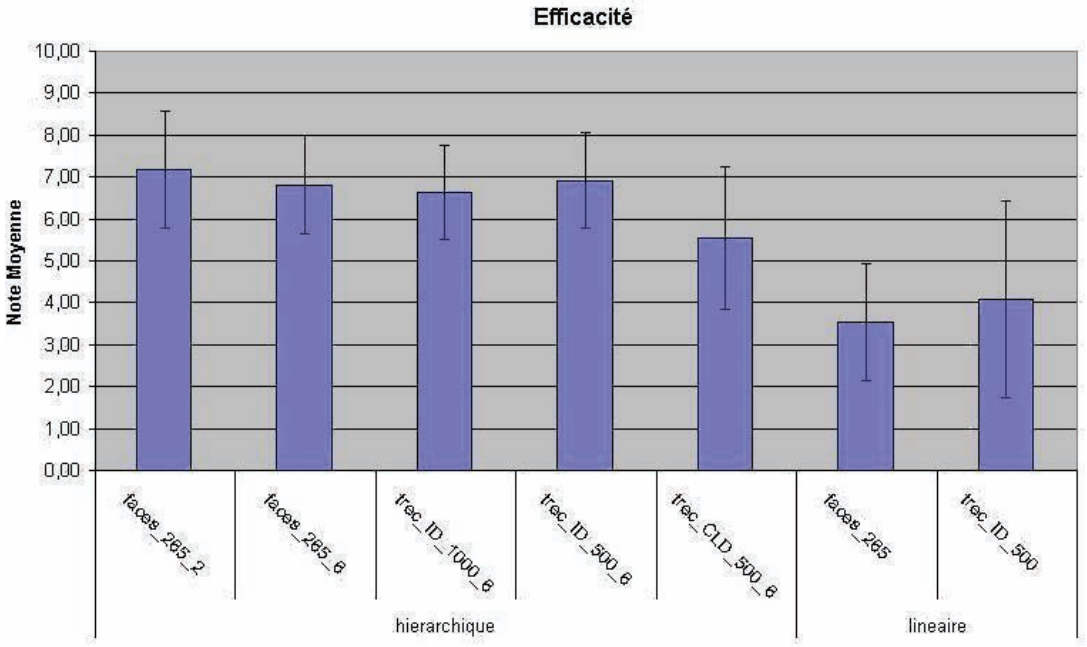


FIG. 5.20: Note d'efficacité moyenne et écart-type attribués par les utilisateurs à l'issue de chacune des sept expériences.

5.5 Discussion

Nous discutons de l'avantage attendu de l'interface linéaire sur l'interface hiérarchique concernant l'évolution du temps de recherche pour des jeux de données de taille plus importante. Nous discutons également des causes du nombre d'échecs lors de l'expérience utilisant l'interface `trec_ID_1000_6_hierarchique`.

5.5.1 Passage à l'échelle

Un argument en faveur de la recherche hiérarchique concerne le temps de recherche et le meilleur passage à l'échelle de ce type d'interface lorsque la taille de la collection augmente. Nous proposons une comparaison des temps de recherche attendus avec chaque type d'interface.

5.5.2 Interface linéaire

Proposition 5.5 (Temps théorique de recherche. Cas linéaire)
Soit une collection d'images de taille n. Le temps théorique de recherche dans la collection d'images est linéaire en fonction de n.

Si le rang auquel on trouve une cible dans l'interface linéaire suit une distribution aléatoire uniforme, le rang moyen est égal à $\frac{n}{2}$. Le temps de recherche moyen sera donc linéaire en fonction de la taille de la collection.

5.5.3 Interface hiérarchique

Pour l'interface hiérarchique, nous proposons une conjecture sur la complexité théorique du processus de recherche. Nous conjecturons que ce temps de recherche dépend du niveau de la hiérarchie à laquelle se trouve la cible ainsi que du nombre de voisins à consulter pour effectuer le choix de navigation à chaque étape. De plus, lorsque la cible est similaire à un grand nombre d'images dans la collection, on considère que le processus de navigation devient une recherche exhaustive dans le voisinage de la cible.

Nous précisons la notion d'*images semblables* dans ce contexte. Soit (S, δ) l'espace métrique composé de l'ensemble S de descripteurs de type D muni de la mesure de dissimilarité δ. Soit ϵ une valeur de dissimilarité en dessous de laquelle deux images sont considérées comme semblables au sens du descripteur utilisé et soit $t \in S$ un descripteur. On définit l'ensemble des images de S semblables à t, noté $Sim(t)$, par

$$Sim(t) = \{u \in S | \delta(t, u) \leq \epsilon\}. \tag{5.7}$$

Temps théorique de recherche. Cas hiérarchique Soit G un graphe de navigation hiérarchique, soit t une cible située au niveau $H(t) \leq h$ dans la hiérarchie des échantillons, soit γ la taille moyenne de $N_u(c), 0 \leq i \leq h, \forall u \in S$. Nous conjecturons que le temps théorique de recherche dans G est $O(H(t).\gamma + Sim(t) \cap S_h)$.

Le premier terme de la complexité correspond à la recherche hiérarchique jusqu'au niveau $H(t)$ où se trouve la cible recherchée. Le temps moyen de ce parcours est proportionnel au nombre de niveaux séparant t de la racine que multiplie le nombre de voisin à consulter à chaque saut.

Le deuxième terme de la complexité permet de tenir compte du temps nécessaire pour distinguer la cible d'un ensemble d'images similaires. Ce temps est proportionnel au nombre de sommets similaires à t se trouvant dans l'ensemble S_h, soit $Sim(t) \cap S_h$.

D'après les données expérimentales dont nous disposons, il nous manque des expériences sur un plus grand nombre de requêtes situées à différentes hauteurs dans pour pouvoir exhiber une corrélation entre les temps effectifs de recherche dans l'interface hiérarchique et une combinaison linéaire des valeurs réelles pour $H(t)$, γ et $Sim(t)$. Une perspective immédiate de ce travail consisterait à conduire des expériences supplémentaires dans ce sens.

5.5.4 Difficulté relatives des cibles

La figure 5.16 indique que les recherches menées sur la collection d'images issues de la collection `trec` à mille éléments conduisent à un nombre d'échecs bien supérieur à celui des recherches menées sur la même collection avec cinq cents éléments.

Les cinq cibles pour la collection à cinq cents éléments étaient différentes de celles choisies pour la collection à mille éléments. Ce choix est justifié par le fait que la première des deux expériences aurait permis un entraînement de l'utilisateur et aurait biaisé les résultats lors de la deuxième recherche. La figure 5.16 indique que les cibles utilisées pour la recherche dans les mille éléments ont été plus difficiles à trouver que celles de la recherche dans la collection à cinq cents éléments. Analysons la nature de ces différences.

La figure 5.21 présente la répartition des valeurs de chaque dimension du descripteur ID pour la collection à mille éléments. Concernant le nombre et la surface des visages, on constate que plus de 60% des images ne contient pas de visages. Lorsque des visages sont présents dans l'image, dans 40% des cas, ils occupent moins de 5% de la surface de l'image. Le nombre de régions présente un pic centré sur la valeur deux. La luminance et la saturation présentent une répartition des valeurs plus homogène. La teinte se répartie essentiellement entre le rouge (valeurs proches de 0) et le bleu (valeurs proches de $0,6$).

Nous proposons de qualifier une image de "difficile à trouver" lorsque les valeurs associées à son descripteur se situent simultanément dans les classes de plus fort effectifs. Auquel cas, aucune des caractéristiques ne peut être utilisée préférentiellement pour guider la navigation de l'utilisateur.

D'après la figure 5.21, une image ne contenant pas de visages, ayant un nombre de régions, une luminance et une saturation moyens et une teinte dominante plutôt rouge correspondrait à une image difficile à trouver selon les critères utilisés pour indexer cette collection d'images.

Le Tableau 5.4 présente les valeurs des descripteurs associés aux cinq images utilisées pour la recherche dans la collection à mille éléments ainsi que les dimensions associées aux indices visuels les plus discriminants.

La figure 5.22 présente le nombre de succès et d'échecs par requête pour les recherches dans la collection à mille éléments. On remarque les mauvaises performances obtenues sur les deux premières requêtes alors que le descripteur semble permettre de discriminer les caractéristiques de ces images.

Concernant ces deux cibles, le mauvais résultat associé doit être attribué essentiellement à un problème d'interprétation du descripteur par l'utilisateur. Pour la cible numéro un, la teinte la plus saturée correspond au disque sombre qui correspond à un violet, saturé et peu lumineux. Certains utilisateurs se sont basé sur la teinte verte du fond de l'image, leur recherche n'a pas pu être menée à terme avec succès. Dans le cas de la deuxième cible, le nombre de visages présents dans l'image n'a pas été correctement interprété par les utilisateurs. Alors que l'indexation se base sur la présence des trois visages des passants, certains utilisateurs ont considéré qu'il n'y avait pas de visages dans la scène. Cette information discriminante n'étant pas utilisée, la recherche est encore plus difficile à accomplir.

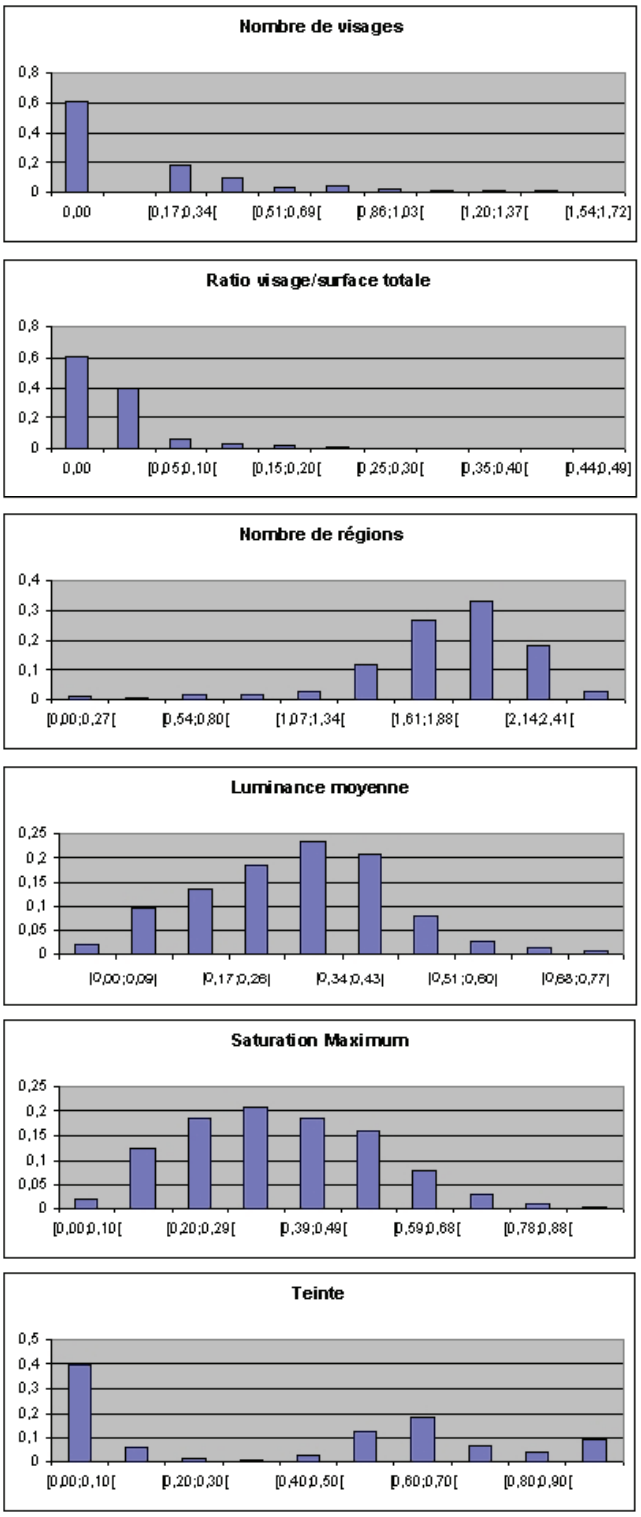
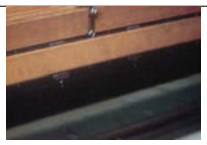


FIG. 5.21: Histogramme des valeurs de descripteur.

	Cible	Descripteur	Dimension discriminante
1		$(0, 0, 1.69, 0.52, 0.25, 0.77)$	luminance, saturation, teinte
2		$(0.70, 0.01, 2.16, 0.40, 0.46, 0.61)$	nombre de visages, nombre de régions, saturation
3		$(1.41, 0.02, 2.27, 0.31, 0.51, 0.07)$	nombre de visages, nombre de régions, saturation
4		$(0.48, 0.17, 2.12, 0.44, 0.55, 0.62)$	nombre de visages, surface visage, saturation, teinte
5		$(0, 0, 1.67, 0.30, 0.46, 0.06)$	luminance, saturation

TAB. 5.4: Cibles utilisées avec la collection d'images-clé à mille éléments.

Le biais lié à l'indexation, explique en partie les plus mauvais résultats obtenus avec ces cibles que ceux obtenus avec les cibles de la collection de taille cinq cents mais il semble que la difficulté des cibles soit aussi en cause. Globalement, la difficultés de ces cibles utilisées pour la recherche dans la collection à cinq cent éléments est plus faible car la présence de visages dans chacune des cinq cibles est une caractéristique discriminante et facilement interprétable.

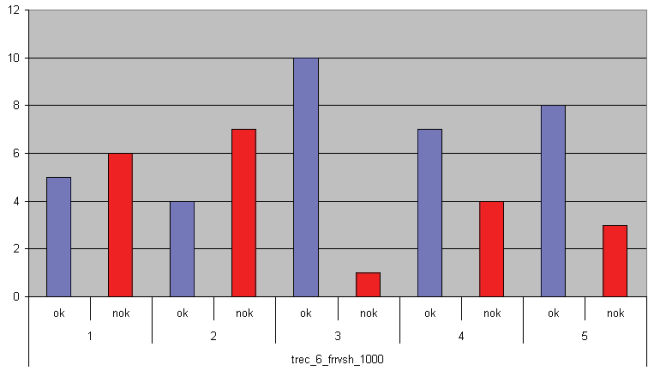
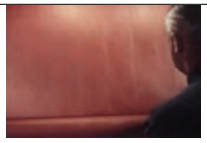


FIG. 5.22: Résultats des recherches pour chaque requête. Collection avec mille éléments.

	Cible	Descripteur	Dimension discriminante
1		$(0.60, 0.03, 2.17, 0.46, 0.34, 0.65)$	nombre de visages, nombre de régions, teinte
2		$(0.48, 0.11, 2.11, 0.40, 0.26, 0.09)$	nombre de visages, surface visage, saturation
3		$(0.30, 0.11, 1.20, 0.41, 0.56, 0)$	nombre de visages, surface visage, nombre de régions, saturation
4		$(0.48, 0.01, 1.91, 0.15, 0.56, 0.05)$	nombre de visages, luminance, saturation
5		$(0.78, 0.10, 1.78, 0.27, 0.06, 0.66)$	nombre de visages, surface visage, luminance, saturation, teinte

TAB. 5.5: Cibles utilisées avec la collection d'images-clé à cinq cents éléments.

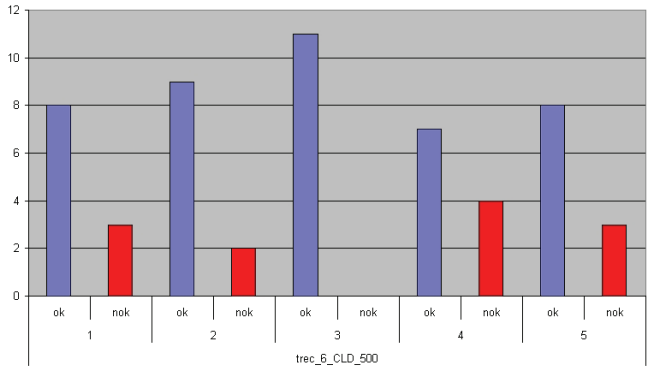


FIG. 5.23: Résultats des recherches pour chaque requête. Collection avec cinq cents éléments.

5.6 Conclusion et perspectives

Nous avons proposé une méthode de recherche visuelle basée sur le paradigme de navigation locale dans un graphe. La structure du graphe est calculée à partir de l'espace métrique composé des descripteurs associés aux images de la collection indexée.

Graphe de navigation L'interface de navigation proposée repose sur une structure de graphe combinant un arbre de recherche construit à partir d'un échantillonage hiérarchique de l'espace métrique. La structure d'arbre permet idéalement d'effectuer une recherche en un temps proportionnel à la hauteur de l'arbre.

La hiérarchie des centres discrets utilisée pour former la hiérarchie d'échantillons offre deux avantages :

- 1°, les propriétés de couverture et de séparation permettent d'offrir, à chaque niveau, un échantillon représentatif de la diversité des images présentes dans la collection. Cela facilite le choix de l'image courante dans les premières étapes de la navigation.
- 2°, la propriété de séparation facilite également la comparaison entre les sommets du voisinage en n'affichant pas deux sommets trop similaires dans la même vue. Cela facilite le choix du voisin le plus similaire à chaque étape de navigation.

L'expérience de recherche dans une collection la collection de visages (collection `face`) , composée d'images indexées très similaires les unes des autres et formant un ensemble de points homogènes dans l'espace de description montre l'efficacité de notre approche dans ce contexte.

Nous avons montré que l'ajout d'arêtes de voisinage entre les sommets de chaque niveau permettait d'éviter les erreurs de navigation liées à la structure d'arbre. Les résultats expérimentaux montrent que l'utilisation de la valeur optimale du rayon de voisinage n'a pas une grande influence sur les performances de navigation par rapport à un réglage plus faible

Descripteur interprétable Notre analyse des résultat concernant la navigabilité dans une collection d'images indexées indique que deux caractéristiques sont requises : l'interprétabilité du descripteur et le pouvoir discriminant du descripteur ou l'homogénéité des points dans l'espace de description :

- Premièrement, le "fossé sémantique" doit être réduit le plus possible par l'utilisation de descripteurs facilement interprétables d'après le contenu de l'image. L'indexation ID produit de meilleurs résultats que l'indexation par CLD sur la même collection d'images et pour le même ensemble de cibles. Ce résultat indique que d'un point de vue pratique, cette caractéristique du descripteur influence la navigabilité de manière significative. L'entraînement de l'utilisateur devrait également permettre de meilleurs résultats que ceux obtenus dans le cadre de l'expérimentation que nous avons conduite dans la mesure où le descripteur serait plus justement et plus rapidement interprété.
- Deuxièmement, l'indexation utilisée doit permettre de répartir de manière homogène l'ensemble des descripteurs associés à des contenus différents dans l'espace de description. Dans le cas contraire, des ensembles d'images sont indistinguables et donc plus difficiles à trouver. Dans ce cas, une nouvelle indexation ou des caractéristiques (dimensions) supplémentaires peuvent être employées pour permettre une meilleure organisation de l'espace de description.

En conclusion, l'indexation ID proposée, bien que plus interprétable et donc plus adaptée que le descripteur CLD pour la recherche dans les images-clé, n'est pas suffisamment discriminante pour une grande partie des images de la collection utilisée. Considérons la collection `trec` avec mille images, soixante-dix images ont simultanément les six valeurs de descripteur situées dans les classes de valeurs représentant plus de 20 pourcent de l'effectif et donc les moins discriminantes. Cela signifie qu'au moins 7 pourcent des images de la collection est difficilement distinguable d'après l'indexation utilisée.

Le descripteur moyen de cette classe d'images à les caractéristiques suivantes :

$$(0.07, 0.01, 1.92, 0.42, 0.37, 0.98),$$

ce qui correspond à une image ne contenant pas de visage, avec une centaine de régions, une luminance moyenne intermédiaire, et la couleur la plus saturée est un rouge faiblement saturé.

La figure 5.24 présente des exemples d'images partageant ces caractéristiques.

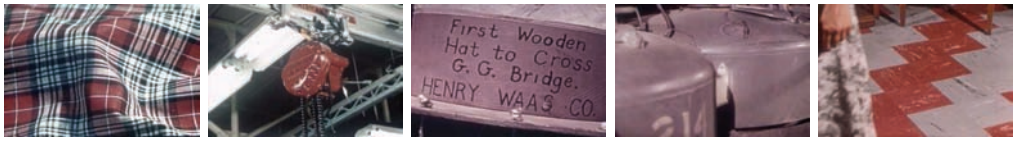


FIG. 5.24: Exemple d'images indistinguables au sens de l'indexation utilisée.

5.6.1 Perspectives

Les expériences conduites nous permettent de tirer des conclusions sur l'intérêt effectif du paramètre de voisinage. Dans des versions futures de l'interface hiérarchique, un voisinage limité à un nombre constant de voisins pourra être utilisé, limitant ainsi la taille de la structure de données et l'affichage sans dégrader les performances de recherche.

Du point de vue de l'indexation, le principe consistant à utiliser des descripteurs interprétables peut être étendu pour enrichir la collection des descripteurs utilisables. Ensuite, une étude du pouvoir discriminant de l'indexation utilisée pour un type de collection d'images peut indiquer si le système d'indexation est suffisant ou s'il doit être enrichi de nouvelles dimensions. Dans notre exemple, on peut imaginer l'ajout d'une septième dimension au descripteur ID constituée du CLD associé à l'image. Ainsi, les images de la figure 5.24 pourraient être distinguées d'avantage par la métrique associée à ce nouveau descripteur sur la base des caractéristiques fréquentielles de l'image.

Par ailleurs, il serait utile d'adapter la métrique utilisée afin de ne pas discriminer deux contenus d'avantage que ne peut le faire l'utilisateur. Imaginons que les seuil de discrimination (valeur d'un JND "Just Noticeable Difference") pour chaque caractéristique utilisée sur chacune des dimensions aient été déterminés. On pourrait alors paramétrer la mesure de dissimilarité pour permettre à une dimension de contribuer à la dissimilarité globale uniquement dans le cas où la différence entre deux images pour cette caractéristique serait perceptuellement distinguable. Ainsi, les images indistinguables au sens des descripteurs utilisés seraient groupées sous le même représentant dans l'arbre d'échantillonage. Un parcours inutile d'arbre serait remplacé par une recherche linéaire dans cet ensemble de sommets.

Enfin, la définition d'un schéma d'indexation général destiné aux images naturelles est très ambitieux et les caractéristiques interprétables utilisables dans ce contexte sont peu nombreuses ([115]). En revanche, l'exploration de contenus visuels plus spécifiques pourrait être réalisée à l'aide de caractéristiques d'indexation adaptées. On peut imaginer une indexation interprétable pour des collections d'imprimés, des collections de textures ou de matériaux. Dans ce cas, l'interface de navigation hiérarchique pour le parcours et la recherche dans ces collections serait adaptée.

Conclusion

Dans cette thèse, nous avons présenté un ensemble d'algorithmes basés sur les graphes pour traiter différents aspects du problème de l'accès aux contenus visuels indexés.

Indexation des documents vidéo Nous avons proposé la détection d'hyper-scènes colorées dans un document vidéo utilisant la signature "X-Ray" d'un plan vidéo et la comparaison de ces signatures par distorsion de codage. Cette méthode permet de caractériser les palettes de couleurs employées dans la réalisation d'un document vidéo.

Cependant, les résultats obtenus avec la méthode de filtrage et de fragmentation du graphe vidéo indiquent que dans certains cas, la méthode proposée ne parvient pas à détecter certaines hyper-scènes. Cela correspond à des documents vidéo dont le contenu couleur varie peu de plan en plan. L'utilisation d'autres paramètres calculés sur les sommets du graphe filtré pourrait permettre une meilleure identification des quasi-cliques. Ces quasi-cliques pouvant s'apparenter à la définition de communautés dans les réseaux sociaux, des métriques de graphe dédiées à l'identification de communautés dans les réseaux sociaux pourraient être utilisées à cet effet [7].

Nous avons également présenté une méthode de détection des scènes de dialogue visuelles basée sur l'analyse des motifs périodiques dans la matrice d'adjacence pondérée du graphe vidéo. Une détection de visages basée sur un modèle de couleur de peau est utilisée pour compléter l'identification de ces scènes. Le modèle de couleur de peau final est obtenu grâce à un processus d'amorçage-filtrage original basé sur la coopération d'un détecteur de visage SVM et d'un modèle de couleur de peau. Cette coopération permet :

- la suppression de nombreux faux-positifs issus du processus de classification par SVM, ce qui améliore la précision des scènes de dialogue détectées,
- l'obtention d'un modèle de couleur de peau entraîné à partir du seul contenu du document vidéo en cours d'analyse, ce qui permet d'améliorer le rappel du détecteur de visages par modèle de couleur de peau.

Recherche par navigation locale Nous avons proposé et expérimenté une approche originale de navigation locale dans un graphe pour la recherche dans les contenus visuels indexés. Cette proposition pose des questions auxquelles nous n'avons pu répondre que partiellement.

La mise en œuvre de l'interface de navigation hiérarchique au travers des expérimentations du chapitre 5 ouvre des perspectives prometteuses. Les expérimentations montrent qu'avec une indexation adaptée à la navigation locale, plus de 90% des recherches dans une collection d'images naturelles a pu être menée à terme. On a également déterminé un seuil concernant le nombre de voisins utiles autour de chaque sommet courant. Dans les versions futures de l'interface de navigation hiérarchique nous envisageons de limiter ce nombre.

Les résultats expérimentaux montrent que le temps nécessaire à l'accomplissement de ces recherches est comparable à celui des mêmes requêtes dans une interface linéaire. Cependant, le modèle de complexité théorique que nous proposons pour l'interface hiérarchique indique , dans le meilleur des cas, un temps de recherche logarithmique en fonction de l"'aspect ratio" de l'espace métrique. Cependant, dans les cas défavorables, le temps de recherche est influencé de manière négative par le nombre d'images semblables à la cible t recherchée, qui force l'utilisateur à effectuer une recherche exhaustive de l'ensemble $Sim(t)$. Des expérimentations supplémentaires sont donc nécessaires pour confirmer ce modèle théorique et mesurer l'évolution du temps de recherche en fonction de la taille de la collection et de la nature des cibles. Il serait également nécessaire de comparer notre approche avec des outils existants dans un scénario d'utilisation réaliste de recherche d'images.

Les évolutions possibles de la recherche par navigation locale dans un graphe sont :

- L'utilisation de graphes navigables tels que les grilles en dimension d associées au descripteur interprétable.
- L'utilisation d'algorithmes probabilistes [153], rapides et décentralisés, pour la création d'une hiérarchie d'échantillons de l'espace métrique dans la construction du NAV-GRAPHE.

Descripteurs de contenus Nous avons introduit la notion d"'interprétabilité" pour les indices visuels utiles dans le cadre de la recherche visuelle par navigation. Ce critère est primordial pour permettre à l'utilisateur d'estimer la valeur d'un descripteur composé de plusieurs indices visuels en observant le contenu de l'image associée. L'expérimentation présentée dans le chapitre 5 montre le gain d'efficacité obtenu avec le descripteur ID pour la recherche dans le graphe de navigation par rapport à l'utilisation du descripteur MPEG-7 CLD.

L'estimation des fonctions psychophysiques (cf. chapitre 5 et annexe A) associées a chaque indice visuel nous semble importante. Pour faciliter le choix de l'image la plus similaire à la cible recherchée au cours de la navigation locale, on peut envisager de montrer des sommets voisins très *contrastés* les uns avec les autres selon les indices visuels qu'ils contiennent. Ce niveau de contraste peut être déterminé par des expériences consistant à déterminer la valeur du seuil de discrimination associé à chaque indice visuelle. Le seuil de discrimination (ou JND pour "Just Noticeable Difference") correspond, pour un stimulus donné, à la valeur de différence qui est juste perceptible. Le niveau de contraste "suffisant" correspond donc à un JND pour chacun des indices visuels du descripteur utilisé. Dans le cas contraire, l'utilisateur ne peut pas déterminer *facilement* le meilleur candidat parmi les voisins du sommet courant.

Nous envisageons d'étudier d'autres indices visuels du point de vue de la perception et d'intégrer les fonctions psychophysiques associées dans une version étendue d'un descripteur interprétable.

Plongement déterministe d'un espace métrique L'algorithme I-PACK que nous proposons permet d'obtenir des niveaux de distorsion comparables avec d'autres algorithmes de MDS de la littérature. Son temps d'exécution est quasi-linéaire quand la dimension doublante de l'espace métrique est constante. L'organisation des images dans le plongement ainsi obtenu permet d'identifier partiellement les groupes d'images similaires.

Cependant, l'extension de la propriété de séparation entre sommets appartenant à des sous-arbres différents peut être envisagée pour améliorer la qualité du plongement. L'ajout

de ce type de contrainte pourrait améliorer sensiblement la distorsion dans le plongement produit. L'étude de la stabilité du plongement quand de nouvelles images sont ajoutées à la collection peut également être étudiée dans le contexte d'un scénario d'utilisation réel.

Dimension doublante d'un espace métrique Dans cette thèse, nous avons souligné l'importance d'une caractéristique associée à un espace métrique qui est sa dimension doublante. Cette mesure est importante car nous avons vu qu'elle influence les complexités des algorithmes de plongement d'un espace métrique en deux dimensions et la construction de graphes pour la navigation locale dans un espace métrique.

Nous avons proposé deux estimations de la dimension doublante d'un espace métrique à partir de l'arbre d'échantillonage associé. Nous avons également présenté un algorithme de génération aléatoire d'espaces métriques de dimension doublante fixée en temps linéaire. Cette génération aléatoire devrait permettre la comparaison des algorithmes traitant des espaces métriques selon le critère de la dimension doublante. Cette caractérisation nous semble importante car elle influence de manière significative la complexité en temps des méthodes de recherche dans les structures de données utilisées pour indexer les données multimédia. La taille de la collection de documents n'est donc pas le seul paramètre à considérer.

Indexation de document vidéo par visualisation orientée-pixels

L'exploration interactive des motifs présents dans la matrice de similarité constitue un outil générique pour l'indexation de la structure d'un document vidéo. Comme nous l'avons vu dans le chapitre 3, les frontières de plans peuvent être détectées en identifiant des blocs carrés le long de la diagonale principale de la matrice de similarité des images consécutives d'un document vidéo [38]. Nous avons vu que les motifs périodiques correspondant aux scènes de dialogue peuvent être identifiés visuellement dans la matrice de similarité des plans vidéo. D'autres motifs d'intérêt peuvent être utilisés dans ce contexte. L'équivalent des "story-units" [176] correspondent à des blocs carrés denses le long de la diagonale principale de la matrice. Ces mêmes carrés denses, resp. pleins, quand ils sont situés hors de la diagonale principale correspondent à des hyper-scènes, resp. à la répétition d'une même séquence vidéo.

La principale difficulté concernant le développement d'un tel outil réside dans la manipulation interactive de matrices de grande taille dont la complexité en mémoire est quadratique en fonction du nombre d'objets à traiter (images ou plans vidéo). Par exemple, la détection des frontières de plans dans un document de vingt minutes nécessiterait le traitement de 30000 images, soit un matrice contenant 9.10^8 entrées, ce qui est prohibitif pour un traitement interactif.

Toutefois, une technique de visualisation basée-pixels [90] peut rendre la manipulation d'une telle matrice envisageable. Cette visualisation consisterait à rafraîchir dynamiquement, la valeur d'une entrée de la matrice à chaque fois que l'affichage de celle-ci doit être recalculé. Cette apparente perte d'efficacité permet cependant de rendre la complexité en temps du processus d'affichage totalement indépendante de la taille de la matrice traitée. En effet, la complexité en temps devenant proportionnelle à la taille de la zone d'affichage. Ainsi, l'affichage d'une matrice de 9.10^8 entrées sur une zone d'affichage de 300×300 pixels nécessiterait de ne recalculer que 9.10^4 valeurs de dissimilarité à chaque rafraîchissement de l'affichage. L'occupation mémoire est linéaire en fonction du nombre d'objets à traiter

dans la mesure où seuls les descripteurs utilisés pour recalculer la valeur de similarité sont chargés en mémoire.

Une telle technique est bien adaptée à l'analyse interactive de la structure des documents vidéo. Comme les valeurs de dissimilarité sont recalculées à la demande, il devient possible de modifier dynamiquement la formule de calcul pour utiliser une combinaisons de descripteurs différents ou bien pour appliquer différentes valeurs de seuillage.

Annexe A

"Maximum likelihood difference scaling"

Dans le domaine de la psychologie expérimentale et de la biologie, un stimulus désigne tout ce qui est de nature à déterminer une excitation chez un organisme vivant : un son, un signal visuel (image ou lumière), une source de chaleur, la sensation de gravité, un événement, un choc électrique, une odeur, etc.

La *fonction psychophysique* associée à un stimulus est une fonction mathématique qui relie l'intensité physique d'un stimulus à la sensation perçue. Cette Annexe présente la méthodologie "Maximum likelihood difference scaling" (MLDS) proposée par Maloney et al. [105] pour estimer la fonction psychophysique associée à la perception d'un stimulus particulier.

La perception qu'a un individu de certains stimuli n'est pas nécessairement proportionnelle aux caractéristiques physiques, mesurables, du stimulus. Par exemple, la sensibilité du système auditif est liée de façon logarithmique à la puissance du signal sonore. Dans la catégorie des sons purs aux fréquences dites moyennes (1000 - 4000 Hz) une augmentation de 10 dB_{SPL} (décibels "Sound Pressure Level"), correspond au doublement du volume sonore perçu. La valeur d'un signal en dB_{SPL} est égale à $10\log_{10}(\frac{P}{P_0})$, où P indique la puissance du signal sonore par unité de surface (en watts par mètre carré) et $P_0 = 10^{-12} w/m^2$ est le niveau de base. Une augmentation de 10 dB_{SPL} correspond donc a une multiplication de la puissance par environ 3.17.

L'expression du niveau sonore en dB_{SPL} permet donc, d'une part, de donner une interprétation du volume sonore en termes de puissance sonore perçue et d'autre part, d'établir une relation linéaire et uniforme (pour les fréquences moyennes), entre le nombre de décibels et le volume sonore perçu.

L'estimation de la fonction psychophysique est utilisée dans le cadre de la compression vidéo [166], et de la compression d'images [91]. Elle permet de mesurer la perception des artefacts de codage en fonction du niveau de compression appliqué aux images.

Nous présentons le modèle de perception ainsi que le protocole expérimental utilisés [105] dans la section A.1.

A.1 Méthode de MLDS

Nous présentons la technique de "maximum likelihood difference scaling" (MLDS) [105] pour estimer la fonction psychophysique associée à la perception d'un stimulus particulier.

La méthodologie MLDS a été utilisée avec succès dans [124] pour évaluer la fonction psychophysique associée à la perception du brillant d'un matériau et dans le cadre de la compression vidéo, pour mesurer la distorsion perçue en fonction de la compression [91].

Dans cette méthode, une séquence ordonnée de quatre stimuli, i, j, k, l, est extraite de l'ensemble des stimuli S. Ces stimuli sont présentés à l'observateur sous forme de deux paires (i, j) et (k, l). La tâche de l'observateur consiste à sélectionner la paire d'éléments qui présente la plus grande différence en terme de la caractéristique mesurée. Si la paire (i, j) est sélectionnée, la valeur $R = 0$ est associée au quadruplet (i, j, k, l), ordonné par valeur, sinon, c'est la valeur $R = 1$ qui lui est associée. Pour une collection de $|S|$ stimuli, il existe $\binom{|S|}{4}$ quadruplets différents. Par exemple, pour $|S| = 7$ stimuli, on peut former 35 quadruplets ordonnés correspondant à 35 tâches.

Le modèle de perception utilisé, un modèle de détection du signal Gaussien à variance égale [65], suppose que chacun des quatre stimuli, i, j, k et l, génère chez l'observateur quatre réponses, notées Ψ_i, Ψ_j, Ψ_k et Ψ_l. Ces valeurs sont inconnues mais sont supposées vérifier l'inégalité suivante :

$$|\Psi_i - \Psi_j| > |\Psi_k - \Psi_l|$$

si et seulement si l'observateur juge que les stimuli de la paire (i, j) sont plus différents que ceux de la paire (k, l).

On considère que la variable de décision utilisée par l'observateur est

$$\Delta = |\Psi_i - \Psi_j| - |\Psi_k - \Psi_l| + \epsilon$$

où ϵ est une variable Gaussienne de moyenne nulle et d'écart-type $\sigma > 0$. Quand l'intervalle $|\Psi_i - \Psi_j| - |\Psi_k - \Psi_l|$ est petit devant ϵ, on s'attend à ce que l'observateur effectue des jugements contradictoires lorsqu'on lui présente les mêmes stimuli.

Quand $\Delta > 0$, l'observateur sélectionne la paire (i, j), sinon, c'est la paire (k, l) qui est choisie. La procédure MLDS permet d'estimer la valeur de la fonction psychophysique aux points correspondant à l'intensité des $|S|$ stimuli, c'est à dire les valeurs $(\Psi_i)_{0 \leq i \leq |S|}$.

Les valeurs correspondant à Ψ_0 et $\Psi_{|S|}$ sont normalisées en fixant leurs valeurs à $\Psi_0 = 0$ et $\Psi_{|S|} = 1$. Les valeurs $(\Psi_i)_{0<i<|S|}$ sont estimées en maximisant la fonction

$$L[\Psi_1, ..\Psi_{|S|-1}, \sigma | R_1, .., R_{\binom{|S|}{4}}] = \prod_{q=1}^{\binom{|S|}{4}} [\Phi_\sigma(\Delta_q)^{1-R_q}(1 - \Phi_\sigma(\Delta_q))^{R_q}]$$

où Φ_σ désigne la distribution normale cumulée de variance σ. D'après [105], la valeur $\sigma = 0.2$ est une valeur typique pour un observateur humain dans le contexte de la perception des couleurs.

$L[\Psi_1, ..\Psi_{|S|-1}, \sigma | R_1, .., R_{\binom{|S|}{4}}]$ est la vraisemblance des paramètres $(\Psi_i)_{1 \leq i \leq |S|-1}, \sigma$ étant donnée les réponses de l'utilisateur-test, notées $(R_i)_{1 \leq i \leq \binom{|S|}{4}}$, lors des expériences. Nous cherchons donc à trouver les valeurs $\Psi_1, ..\Psi_{|S|-1}$ qui maximisent la fonction objectif

$$L[\Psi_1, ..\Psi_{|S|-1}, \sigma | R_1, .., R_{\binom{|S|}{4}}].$$

C'est un problème d'optimisation qui peut être résolu par programmation linéaire en utilisant l'algorithme du simplexe et une technique de recuit simulé [132].

Annexe B

Approximation de la dimension doublante

Comme il n'est pas possible de calculer la dimension doublante d'un espace métrique en temps polynomial [72], nous sommes intéressés par son approximation. Cette annexe présente deux résultats sur l'approximation de la dimension doublante d'un espace métrique à partir de l'arbre d'échantillonage associé.

Le Lemme B.1 donne une borne supérieure sur le degré sortant maximum de l'arbre d'échantillonage T d'un espace métrique (S, δ). Ce résultat permet de donner une borne inférieure sur la dimension doublante d'un espace métrique (S, δ) étant donné le degré sortant maximum d'un arbre d'échantillonage associé à (S, δ).

Proposition B.1 (Borne supérieure du degré de l'arbre d'échantillonage)
Soit dd la dimension doublante d'un espace métrique (S, δ). Soit T un arbre d'échantillonage associé à (S, δ). Le degré maximal de T, noté α, est borné par 2^{2dd}.

Preuve de la proposition B.1 : Soit l'ensemble de sommets S et soit $u \in S_i$. On définit $\mathcal{C}$, l'ensemble des sommets de S_{i-1} couverts par la boule de rayon 2^i centrée en u : $\mathcal{C} = S_{i-1} \cap B_u(2^i)$. Puisque les fils de u au niveau S_{i-1} sont inclus dans $\mathcal{C}$, on a : $|C_{i-1}(u)| \subseteq \mathcal{C}$.

D'après la définition de la dimension doublante, $B_u(2^i)$ peut être inclus dans $\ell \leq 2^{dd}$ boules de rayon 2^{i-1} centrées en $u_1, ..., u_\ell \in S$. Notons $\mathcal{C}_O = \{u_1, ..., u_\ell\}$ et $B = \bigcup_{u_j \in \mathcal{C}_O} B_{u_j}(2^{i-1}) \supseteq B_u(2^i) \supseteq \mathcal{C}$. Soit u_1 le sommet de $\mathcal{C}_O$ qui couvre le plus grand nombre de sommets de $\mathcal{C}$. Soit $k = \max\limits_{u_j \in \mathcal{C}_O} |\mathcal{C} \cap B_{u_j}(2^{i-1})| = |\mathcal{C} \cap B_{u_1}(2^{i-1})|$.

Chaque point de $\mathcal{C}$ est couvert par au plus 2^{dd} boules $B_{u_j}(2^{i-1})$ et au moins 1 boule $B_{u_j}(2^{i-1})$.

On a donc :

$$|\mathcal{C}| \leq \sum_{u_j \in \mathcal{C}_O} |\mathcal{C} \cap B_{u_j}(2^{i-1})| \leq \ell.|\mathcal{C} \cap B_{u_1}(2^{i-1})| \leq 2^{dd}.k.$$

Déterminons k. Considérons $X = \mathcal{C} \cap B_{u_1}(2^{i-1})$. Par définition, l'ensemble $B_{u_1}(2^{i-1})$, et donc X, peut être couvert par au plus $\ell' \leq 2^{dd}$ boules de rayon 2^{i-2}. Or X est une $2^{i-1}-$séparation de S. Il en résulte que $\forall B'_{u_j}(2^{i-2}), |B'_{u_j}(2^{i-2}) \cap X| \leq 1$. En effet, si $x_1, x_2 \in B'_{u_j}(2^{i-2})$, alors $\delta(x_1, x_2) \leq 2^{i-1}$ donc x_1 et x_2 ne peuvent pas appartenir simultanément à X :

$$k = |X| \leq \sum_{j=1}^{\ell'} |B'_{u_j}(2^{i-2}) \cap X| \leq \ell' \leq 2^{dd}.$$

On peut conclure que $\forall u \in S, 0 \leq i \leq h$, $|C_{i-1}(u)| \leq |\mathcal{C}| \leq 2^{dd}.2^{dd} = 2^{2dd}$. Le degré de T, noté α est borné par 2^{2dd}.

A l'inverse, soit T un arbre d'échantillonage et $\alpha = \max_{u \in T} deg^+(u)$, alors on obtient la borne inférieure suivante de la dimension doublante de l'espace métrique associé (S, δ) :

$$\alpha \leq 2^{2dd} \Rightarrow \frac{\log_2 \alpha}{2} \leq dd.$$

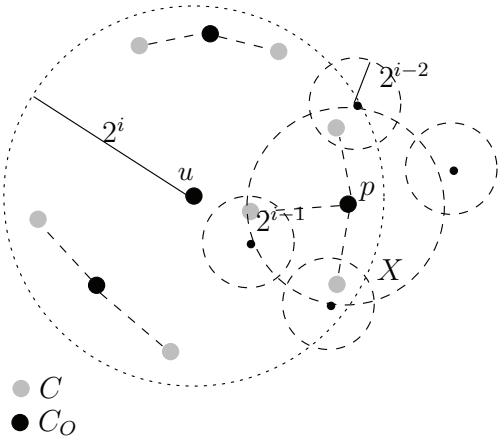


FIG. B.1: Couverture dans T par rapport à la couverture optimale.

Approximation de la dimension doublante La proposition B.2 permet de donner une approximation de la dimension doublante d'un espace métrique (S, δ).

Soit $u_j \in S_i$, $\mathcal{C}_G^{u_j}$ est l'ensemble de points défini par :

$$\mathcal{C}_G^{u_j} = S_{i-2} \cap B_{u_j}(3.2^{i-1}).$$

Proposition B.2 (Approximation de la dimension doublante)
Soit (S, δ) un espace métrique, soit $\{S_i\}_{0 \leq i \leq h}$ une hiérarchie des centres discrets de S et soit $c = \max_{\forall u_j \in S_i, 0 \leq i \leq h} |\mathcal{C}_G^{u_j}|$. La dimension doublante, notée dd, de S vérifie la relation :

$$\frac{\log_2 c}{3} \leq dd \leq \log_2 c.$$

Cette approximation peut être calculée par l'algorithme **`dd_approx`** *en temps $O(2^{2dd} n \log A)$.*

Preuve du proposition B.2 : Par une preuve similaire à celle du Lemme B.1, on peut encadrer $|\mathcal{C}_G^{u_j}|$ en fonction de la taille de la couverture optimale de $B_{u_j}(2^i)$, notée $|\mathcal{C}_O|$. On montre que pour $c = \max_{\forall u_j \in S_i, 0 \leq i \leq h} |\mathcal{C}_G^{u_j}|$ on a :

$$|\mathcal{C}_O| \leq 2^{dd} \leq c \leq k|\mathcal{C}_O| \leq 2^{3dd}.$$

Ce qui implique : $dd \leq \log_2(c)$ et $\frac{\log(c)}{3} \leq dd$.

A partir de l'arbre d'échantillonage T, on peut obtenir une approximation de la dimension doublante des données en déterminant la valeur de $c = \max_{\forall u \in T, 2 \leq i \leq \ell_u} |\mathcal{C}_G^u|$. Cette valeur peut être déterminée à l'aide de plusieurs parcours de T.

L'algorithme `dd_approx` calcule la valeur de c en effectuant un parcours des sommets de T et en calculant pour chaque sommet u, la valeur de $|\mathcal{C}_G^u|$. Le nombre maximum de sommets dans cet ensemble à chaque étape du calcul est le nombre de descendants situés deux niveaux en dessous du sommet courant, soit 2^{2dd}. La complexité en temps de cet algorithme est donc $O(2^{2dd} n \log A)$. □

Données: L'arbre T associé à l'espace métrique (S, δ)
max $\leftarrow 0$
Pour $u \in S_2$ **Faire**
  Pour niveau i de 2 à ℓ_u **Faire**
    $C = \bigcup_{w \in C_{i-1}(u)} C_{i-2}(w)$
    courant $\leftarrow |\{v | \delta(u, v) \leq 3.2^i\}|$
    Si courant $>$ max **Alors**
      max $\leftarrow$ courant
    FinSi
  FinPour
FinPour
Retour $\log_2(\text{max})$

Algorithme B.1: Algorithme `dd_approx`

Annexe C

Génération aléatoire d'espaces métriques

Pour comparer les performances de l'algorithme I-Pack et les autres algorithmes de MDS (cf. chapitre 4), nous avons besoin d'utiliser différents jeux de données de dimension doublante donnée. Dans l'annexe B, nous avons montré une relation entre la dimension doublante, dd, d'un espace métrique (S, δ) et le degré maximum α de l'arbre d'échantillonage T de cet espace métrique. Le degré de T est borné par 2^{2dd}.

Dans cette annexe, nous montrons que nous sommes capables de construire un espace métrique aléatoire de taille n, de dimension doublante donnée en temps linéaire.

Notre méthode se décompose en deux phases :

- Génération aléatoire et uniforme d'un arbre T de degré maximal α, avec $n + O(\sqrt{n})$ feuilles et une distribution des degrés donnée $\Pi = (p_0, .., p_i)$, où $p_i = \mathbb{P}(deg^+(u) = i)$. La technique utilisée est celle des arbres de Galton-Watson, car tous les arbres générés d'après une distribution des degrés donnée sont équiprobables. Cette phase est décrite dans la section C.1.
- Pondération des arêtes de l'arbre et calcul des distances entre chaque paire de feuille u, v correspondant à la distance entre u et v dans l'espace métrique associé, c'est-à-dire $\delta(u, v) = d_T(u, v)$. C'est la phase de définition de (S, δ). Cette phase est présentée dans la section C.2.

Dans la section C.2, nous montrons que la dimension doublante de (S, δ) ainsi généré est égale à $\log_2(2\alpha - 1)$. Ainsi, nous pouvons générer une matrice de distance correspondant à un espace métrique de dimension doublante dd fixée en prenant une distribution Π telle que $\forall i > \alpha$, $p_i = 0$ et $p_\alpha = \Omega(\sqrt{\frac{\log n}{n}})$ pour $\alpha = 2^{dd-1} + \frac{1}{2}$.

Un intérêt supplémentaire de cette méthode est le codage compact de l'espace métrique généré. En effet, les métriques produites sont représentées par un arbre pouvant être codé avec $O(n \log A)$ bits. Il n'est pas nécessaire de stocker une matrice de distance qui nécessiterait $\Omega(n^2 \log A)$ bits. Dans ce cas, le temps de calcul des distances dans l'arbre est $O(\log A)$ contre $O(1)$ avec une matrice.

C.1 Génération aléatoire d'arbres

Dans l'annexe B, nous montrons que la dimension doublante d'un espace métrique est liée au degré maximum dans l'arbre d'échantillonage associé. La caractéristique de l'espace métrique que l'on souhaite produire est la dimension doublante. Ainsi, en choisissant le

degré maximum, noté α de l'arbre généré et en associant des poids particuliers aux arêtes de l'arbre, la distance dans l'arbre entre toute paire de feuille correspond aux distances dans l'espace métrique généré.

Par ailleurs, nous souhaitons que les arbres générés T aient une distribution des degrés similaire à celle observée sur des jeux de données réels.

C.1.1 Notations

Nous introduisons les notations suivantes pour désigner différentes familles d'arbres aléatoires :

- $\mathcal{T}_N$: arbres à N sommets.
- $\mathcal{T}_{N,\Pi}$: arbres à N sommets, dont la distribution des degrés est Π : $D_i(T) = \frac{p_i}{|T|}$. $D_i(T)$ est le nombre de sommets de T dont le nombre de descendants est égal à T. $D_0(T)$ correspond au nombre de feuilles dans T.
- $\mathcal{T}_{N,\Pi,\alpha}$: arbres à N sommets, de degré maximum α et dont la distribution des degrés est Π : $D_i(T) \propto \frac{p_i}{|T|}$, avec $p_i = 0,\ \forall i > \alpha$.

Nous introduisons la relation de ϵ-similarité entre deux distributions Π et Π'. La distribution Π' est "$\epsilon-$similaire" à Π si :

$$\forall i \in [0, \alpha],\ |D_i(T) - \mathbb{E}(D_i(T))| < \epsilon.$$

L'ensemble des arbres $\epsilon-$similaires à $\mathcal{T}_{N,\Pi,\alpha}$ est :

$$[\mathcal{T}_{N,\Pi,\alpha}]^{\epsilon} = \bigcup_{\Pi'\epsilon-\text{similaire à}\Pi} \mathcal{T}_{N,\Pi',\alpha}.$$

Définition 66 (Conditions sur Π)
Dans cette annexe, nous considérons les distributions Π *telles que :*

- $p_0 = \Theta(1)$,
- $p_i = 0, \forall i > \alpha$,
- $\forall i$ *tel que* $p_i > 0$, *on a* $p_i = \Omega(\frac{\log N}{N})$.

C.1.2 Processus de Galton-Watson

On propose d'utiliser une méthode de génération aléatoire basée sur la simulation d'un processus de branchement encore appelé processus de Galton-Watson. Ce type de processus est utilisé pour modéliser une population dans laquelle chaque individu appartenant à une génération g_n produit un nombre d'individus appartenant à la génération g_{n+1} selon une distribution de probabilité Π identique pour chaque individu. Cette approche est due à Galton et Watson qui l'utilisèrent en 1874 pour modéliser la probabilité d'extinction des noms de famille aristocratiques [167].

Nous utilisons cette approche car elle permet d'effectuer une génération aléatoire et uniforme dans l'univers $\mathcal{T}_{N,\Pi}$.

On se donne une suite $\Pi = (p_k)_{k\in\mathbb{N}}$ de nombre positifs ou nuls, tels que

$$\sum_{k=0}^{+\infty} p_k = 1.$$

Chaque individu, au cours du processus de branchement, aura une probabilité p_k d'avoir k fils. La suite $(Z_n)_{n\in\mathbb{N}}$ de variables aléatoire, modélise le nombre d'individus de la population à la génération n.

Si l'on suppose que le processus démarre à partir d'un seul individu, à la génération 0 ($Z_0 = 1$), la loi de probabilité du nombre d'individus à la génération suivante sera décrite par les nombre p_k :

$$\mathbb{P}(Z_1 = k) = p_k, \forall k \in \mathbb{N}.$$

Les individus de la génération 1 deviennent à leur tour parents selon la même loi de probabilité. Si l'on a k individus à la génération 1, la loi de probabilité de Z_2 sera alors la même que celle de la somme de k copies indépendantes de la variable aléatoire Z_1.

La famille d'arbre générée par le processus de branchement, notée $\mathcal{T}_\Pi$, est incluse dans la famille plus vaste, des arbres, notée $\mathcal{T}$.

C.1.3 Méthode de génération

Le processus de branchement s'exprime très simplement par un algorithme à rejet pour la génération d'arbres dont le nombre de feuille, $D_0(T)$ et le degré maximum α sont fixés.

L'algorithme `generer_arbre` renvoie des arbres de la famille $\mathcal{T}_{N',\Pi',\alpha}$ dont le nombre de sommets N' est compris entre $(1-\epsilon')N$ et $(1+\epsilon')N$, dont la distribution des degrés est ϵ-similaire à Π et dont le degré maximum est égal à α.

L'algorithme `generer_arbre` fait appel aux fonctions et procédures suivantes :

- La fonction `choisir_degree` renvoie une valeur suivant la distribution Π.
- La procédure `ajouter_enfant(`T`,`i`,`F`)`, ajoute F fils au i-ème sommet dans l'ordre d'un parcours en largeur de T.
- La fonction `prochain_bfs(`T`,`i`)`, renvoie le prochain sommet de T en partant du sommet i-ème selon l'ordre du parcours en largeur de T.

```
Données: Π, N, ε'
T ← ({r}, ∅)
i ← 0
somme_aretes ← 0
Faire
  F ← choisir_degre(Π)
  ajouter_enfant(T, i, F)
  i ← prochain_bfs(T, i)
  somme_aretes ← somme_aretes +F
Jusqu_à somme_aretes = i Ou |T| > (1 + ε')N
Si D_α(T) = 0 Ou |T| > (1 + ε')N Ou |T| < (1 − ε')N Alors
  T ← generer_arbre(Π,N,ε')
FinSi
Retour T
```

Algorithme C.1: Algorithme `generer_arbre`

Sous réserve des conditions sur Π (cf. définition 66) tout arbre généré par l'algorithme `generer_arbre` appartient à $[\mathcal{T}_{N,\Pi,\alpha}]^\epsilon$ avec grande probabilité (cf. corollaire plus loin).

Le temps d'exécution de la boucle principale de l'algorithme C.1 dépend de la probabilité d'extinction de la population engendrée par le processus de branchement. Si la probabilité d'extinction est strictement inférieure à 1, cela signifie que la boucle principale de l'algorithme C.1 peut ne pas s'arrêter et construire un arbre de taille infinie. Cette probabilité d'extinction dépend de la distribution de probabilité Π utilisée.

D'après la théorie des processus de branchement, la probabilité d'extinction de la population est égale à 1 si et seulement si l'espérance du nombre de fils d'un individu est égale à 1, i.e. $\mathbb{E}(Z_1) = 1$. Dans ce cas, le processus est qualifié de critique [74].

Définition 67 (Conditions de Galton-Watson)
La distribution de probabilités utilisée dans l'algorithme **`generer_arbre`** *doit respecter les conditions suivantes :*

1. $\mathbb{P}(Z_1 = k) = p_k$,
2. $\mathbb{E}(Z_1) = 1$.

La première condition permet d'obtenir une distribution des degrés proche de celle des arbres d'échantillonage associés aux jeux de données réels. La dernière condition, garantit l'extinction de la population et donc l'obtention d'arbres de taille finie.

Si la distribution de probabilité utilisée conduit à un processus de branchement critique, cela garantit l'arrêt de la boucle principale de l'algorithme C.1 avec probabilité 1.

La complexité en temps de l'algorithme C.1 dépend du nombre de rejets nécessaires pour obtenir un arbre dont le nombre de sommets appartient à l'intervalle $[(1-\epsilon')N, (1+\epsilon')N]$.

C.1.4 Nombre de feuilles et distribution des degrés

On montre que le nombre de feuilles n dans un arbre $T \in \mathcal{T}_{N,\Pi,\alpha}$ est proportionnel au nombre de sommets dans T si p_0 est une constante. Cela signifie que la complexité en temps de la génération d'un arbre à n feuilles est la même que celle d'un arbre à N sommets.

Proposition C.1 (Nombre de feuilles dans $|T|$)
Soit T un arbre généré par l'algorithme **`generer_arbre`** *avec $\Pi = (p_k)_{k\in\mathbb{N}}$ et p_0 est une constante. Soit N le nombre de sommets de T et n le nombre de feuilles de T. Avec probabilité $1 - O(\frac{1}{N})$, on a :*

$$(1-\epsilon)p_0 N \leq n \leq (1+\epsilon)p_0 N \tag{C.1}$$

pour tout ϵ tel que $\sqrt{\frac{3\ln N}{p_0 N}} \leq \epsilon < 1$.

Preuve de la proposition C.1 : Soit F_i la variable aléatoire correspondant au nombre de descendants du sommet u_i. On a $\mathbb{P}(F_i = k) = p_k$.

Soit la variable X_i telle que

$$X_i = \begin{cases} 1 & \text{si } F_i = 0, \\ 0 & \text{sinon.} \end{cases} \tag{C.2}$$

Supposons que le processus de branchement se termine en produisant un arbre T de taille N. Le nombre de feuilles de T est $X = \sum_{i=1}^{N} X_i = n$ et l'espérance de X est

$$\mathbb{E}(X) = \mathbb{E}(\sum_{i=1}^{N} X_i).$$

Par linéarité de l'espérance, on a

$$\mathbb{E}(X) = \sum_{i=1}^{N} \mathbb{E}(X_i).$$

Or, $\mathbb{E}(X_i) = p_0$, donc

$$\mathbb{E}(X) = p_0 N.$$

En utilisant la borne de Chernoff générale [120], on a :

$$\mathbb{P}(|\mathbb{E}(X) - X| > \epsilon.\mathbb{E}(X)) \leq 2.e^{-\epsilon^2 \mathbb{E}(X)/3} = A(\epsilon). \tag{C.3}$$

On désire avoir $A(\epsilon) \leq \frac{1}{N} = e^{-\ln N}$. Ainsi, ϵ est choisi de telle manière que :

$$\frac{\epsilon^2 p_0 N}{3} \geq \ln N \Rightarrow \epsilon \geq \sqrt{\frac{3 \ln N}{p_0 N}}. \tag{C.4}$$

On a donc $(1-\epsilon)p_0 N \leq n \leq (1+\epsilon)p_0 N$ pour tout $\sqrt{\frac{3\ln N}{p_0 N}} \leq \epsilon \leq 1$ avec grande probabilité. □

D'après la proposition C.1, le temps nécessaire pour générer une arbre T avec n feuilles est proportionnel au temps nécessaire pour générer un arbre de taille N.

En reprenant le même schéma de preuve, nous obtenons un résultat plus général sur la distribution des degrés :
Corrolaire : Si $\forall i$ tel que $p_i > 0$, on a $p_i > \frac{3\ln N}{N}$, alors la distribution Π' des degrés de l'arbre généré T est $\epsilon-$similaire à Π.

C.1.5 Temps de génération

Nous estimons le nombre de rejets nécessaires ainsi que la taille moyenne des arbres rejetés afin d'évaluer la complexité en temps de l'algorithme de génération `generer_arbre`.

Proposition C.2 (Temps de génération de T)
Soit T l'arbre généré avec $\Pi = (p_k)_{k\in\mathbb{N}}$ en respectant les conditions de Galton-Watson et les conditons sur Π. Soit $\epsilon' \in]0;1[$ une constante et soit $\sigma = \Omega(1)$ la variance de Π. En moyenne, le temps de génération d'un arbre $T \in [\mathcal{T}_{N,\Pi,\alpha}]^\epsilon$ avec $(1-\epsilon')N \leq |T| \leq (1+\epsilon')N$ sommets est en $O(N)$.

Preuve de la proposition C.2 : Soit $|T|$ la taille de l'arbre en cours de construction. A chaque instant du processus de génération aléatoire d'un arbre T de taille $|T|$, on se trouve dans exactement un des cas suivants :

- Cas A, *l'arbre T est trop petit* : $|T| < (1-\epsilon')N$.
- Cas B, *l'arbre T a la bonne taille* : $(1-\epsilon')N \leq |T| \leq (1+\epsilon')N$.
- Cas C, *l'arbre T est trop grand, arrêt de la génération et retour en phase A* : $|T| > (1+\epsilon')$.

Dans les cas A et B, l'arbre a fini sa construction, contrairement au cas C. Soit T_{fin} l'arbre en sortie de l'algorithme `generer_arbre`, cet arbre est de type B. Le temps de génération de T_{fin} correspond au temps de génération d'arbres de type A et C avant de construire un arbre de type B.

Les probabilités pour que l'arbre en cours de construction se termine en phase A,B ou C sont respectivement :

$$p_A = \mathbb{P}(|T| < (1-\epsilon')N) = \sum_{i=1}^{(1-\epsilon')N-1} \mathbb{P}(|T| = i) \quad \text{(C.5)}$$

$$p_B = \mathbb{P}((1-\epsilon')N \leq |T| \leq (1+\epsilon')N) = \sum_{i=(1-\epsilon')N}^{(1+\epsilon')N} \mathbb{P}(|T| = i) \quad \text{(C.6)}$$

$$p_C = \mathbb{P}(|T| > (1+\epsilon')N + 1) \quad \text{(C.7)}$$

Soit X_i la taille du i^{eme} arbre généré dans la boucle principale de l'algorithme `generer_arbre`. X est une variable aléatoire représentant le temps total, $X = \sum_{i=1}^{\ell} X_i$ où ℓ est le nombre d'arbres à générer avant l'obtention d'un arbre de type B. ℓ est une variable aléatoire qui suit une loi géométrique de paramètre p_B . Son espérance est donc $\frac{1}{p_B}$. Conditionant X_i par le type d'arbre construit, nous avons :

$$\mathbb{E}(X_i|A) < (1-\epsilon')N \quad \text{(C.8)}$$

$$\mathbb{E}(X_i|B) \leq (1+\epsilon')N \quad \text{(C.9)}$$

$$\mathbb{E}(X_i|C) = (1+\epsilon')N + 1 \quad \text{(C.10)}$$

Ainsi,

$$\mathbb{E}(X) = \sum^{\frac{1}{p_B}} p_A.\mathbb{E}(X|A) + p_B.\mathbb{E}(X|B) + p_C.\mathbb{E}(X|C) \quad \text{(C.11)}$$

L'arbre courant étant construit selon un processus de Galton-Watson, nous avons [108] :

$$\mathbb{P}(|T| = N) \simeq \frac{N^{-\frac{3}{2}}}{\sqrt{2\pi}\sigma} \quad \text{(C.12)}$$

où σ désigne la variance de la distribution des degrés de T.

Calculant p_B avec $\epsilon' \in]0,1[$ et $\sigma = \Omega(1)$, on obtient $\frac{1}{pB} = \Theta(\sqrt{N})$. En moyenne, on doit donc construire $\sqrt{N}$ arbres avant d'obtenir un arbre de type B.

Calculant $(p_A.\mathbb{E}(X|A) + p_B.\mathbb{E}(X|B) + p_C.\mathbb{E}(X|C))$ avec $\epsilon' \in]0,1[$ et $\sigma = \Omega(1)$, on obtient $p_A.\mathbb{E}(X|A) + p_B.\mathbb{E}(X|B) + p_C.\mathbb{E}(X|C) = \Theta(\sqrt{N})$. En moyenne, la taille des arbres rejetés est $\sqrt{N}$.

On a donc $\mathbb{E}(X) = O(N)$. Le temps de génération d'un arbre $|T|$ de taille N est linéaire. □

Pour générer un arbre T avec un nombre de feuille n, on générera des arbres de taille $p_0.n$ en temps linéaire en utilisant l'algorithme `generer_arbre`. Les arbres tels que

$$(1-\epsilon')p_0N \leq D_0(T) \leq (1+\epsilon')p_0N$$

sont conservés et leurs arêtes sont étiquetées pour générer les distances dans l'espace métrique associé. La section suivante détaille l'étiquetage des arêtes.

C.2 Étiquetage et distances

Dans cette partie, nous présentons la méthode de génération d'un espace métrique, comportant des fragments, avec une dimension doublante fixée.

Dans la section 1.5.3 nous expliquons qu'un espace métrique peut être couvert par un arbre d'échantillonage dont les feuilles correspondent aux éléments de l'espace métrique. Nous présentons ici la fonction de distance entre les sommets d'un arbre $T \in \mathcal{T}_{\Pi,n,\alpha}$ qui permet de créer la matrice de distances entre toute paire de feuilles de T.

L'algorithme C.2 a trois entrées : le nombre d'éléments de l'espace métrique souhaité, noté n, la taille des fragments, notée k et la distribution de probabilités Π. Il renvoie la matrice de distances à n entrées, correspondant à un espace métrique de dimension doublante $\log_2(\alpha)$.

Données: n, k, Π
$T \leftarrow$ generer_arbre(Π, $p_0 n$, α)
prolonger_feuilles (T)
ajouter_fragments(T, k)
M = fabriquer_matrice(T)
Retour M

Algorithme C.2: Algorithme generer_dd

La procédure `prolonger_feuilles` prolonge chaque feuille de T de sorte qu'elles aient toute la même profondeur. La procédure `ajouter_fragments` ajoute k fils à chacune des feuilles de $\mathcal{T}$.

La figure C.1 illustre les étapes 1, 2 et 3 de l'algorithme de `generer_dd`.

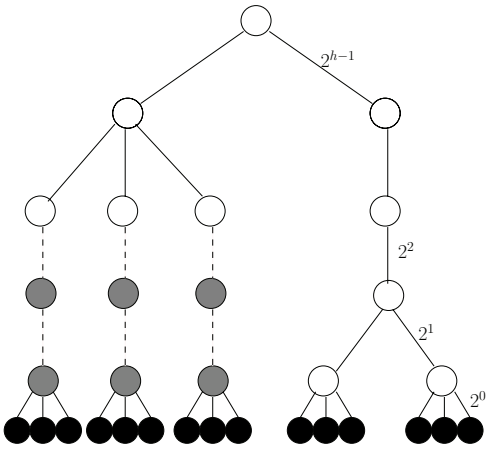


FIG. C.1: Génération d'espace métrique avec fragments. Un arbre aléatoire de taille N est créé (sommets blancs). Les feuilles sont prolongées jusqu'à la hauteur h (sommets gris). $k = 3$ feuilles sont ajoutées aux feuilles existantes (sommets noirs).

Définition 68 (Distance dans T et espace métrique (S, δ))
Soit $T = (V, E)$ un arbre, soient $u, v \in V$. Pour toute arête $e = (u, v) \in E$ telle que $prof(u) < prof(v)$, on définit le niveau de l'arête $l(e) = h(T) - prof(v)$. La valuation des arêtes de T, est $f_E : E \to \mathbb{R}$, $f_E(e) = 2^{l(e)}$. L'espace métrique (S, δ) est définit par :

– $S = \{v \in T | deg^+(v) = 0\}$,
– $\forall u, v \in S$, $\delta(u, v) = d_T(u, v)$ *où* $d_T(u, v)$ *est la distance entre* u *et* v *dans l'arbre* T.

La fonction `fabriquer_matrice` renvoie une matrice de distances correspondant à la distance d_T entre chaque paire de feuilles de T. D'après la définition de (S, δ), il n'est cependant pas nécessaire de stocker une matrice pour calculer les distances dans (S, δ) car l'arbre permet d'effectuer le calcul à la volée.

Montrons maintenant que nous maitrisons la dimension doublante de l'espace métrique généré.

Proposition C.3 (Dimension doublante de l'espace métrique généré)
Soit un arbre $T \in \mathcal{T}_{N,\Pi,\alpha}$. *Soit* S_0, *l'ensemble des feuilles de* T *et soit* (S_0, δ) *l'espace métrique associé aux feuilles de* T. (S_0, δ) *est tel que :*

$$dd = \log_2(2\alpha - 1) \Leftrightarrow 2^{dd} = 2\alpha - 1 \Leftrightarrow \alpha = 2^{dd-1} + \frac{1}{2}.$$

Preuve de la proposition C.3 : Séparons la preuve en deux parties.

Montrons que $dd \leq \log(2\alpha - 1)$: $u^{(0)}$ est la représentation du point u dans l'arbre T de degré maximum α. Par soucis de lisibilité, nous notons $u^{(0)} = u$.

Montrons que $\forall i \in [1, \lceil \log A \rceil]$, $B_u(2^i)$ peut être couvert par $2\alpha - 1$ boules de rayon 2^{i-1}. Rappelons que S_{i-2} est une $2^{i-1}-$domination de S.

Considérons $C_{i-2}(v)$, les fils de v de niveau $i-2$. Par définition, $|C_{i-2}(v)| \leq \alpha$. Nous montrons que $C_{i-2}(v)$ $2^{i-1}-$domine les sommets de $B_u(2^i)$.

Soit $x \in T_{v^{(i-1)}}$ et $y \in T \backslash T_{v^{(i-1)}}$. Par construction de l'arbre, $\delta(x, y) \geq \delta(v, y) \geq 2^i$ sauf pour $x = v$, $y = P_{(i)}(v)$ et $\ell_v \geq i$ (dans ce dernier cas, $\delta(x, y) = 0$).

Soit $v' \in C_{i-2}(v)$, $C_{i-2}(v)$ $2^{i-1}-$domine $T_{v^{(i-1)}}$. Tout sommet de $T_{v'}$ est à distance inférieure à 2^{i-1} de v'.

– Cas 1 : $\ell_u < i - 1 \Rightarrow \delta(u, v) \geq 2^{i-1}$ et $\delta(u, y) = \delta(u, v) + \delta(v, y) > 2^i$. $C_{i-2}(v)$ $2^{i-1}-$domine $B_u(2^i)$.
– Cas 2 $(u = v)$: $\ell_u = i - 1 \Rightarrow v = u^{(i-1)}, \delta(u, v) = 0$. Or $\delta(v, w) = 2^i = \delta(u, w)$. $\{w\} \cup C_{i-2}(v)$ $2^{i-1}-$domine $B_u(2^i)$.
– Cas 3 $(u = v = w)$: $\ell_u > i - 1 \Rightarrow \delta(u, w) = 0$.

Dans $T \backslash T_v^{i-1}$, seuls les éléments de $C_{i-1}(w)$ sont à distance $\leq 2^i$ de w, donc de u. $C_{i-1}(w) \cup C_{i-2}(w)$ $2^{i-1}-$domine $B_u(2^i)$ or $u \in C_{i-1}(w) \cap C_{i-2}(w)$ donc $|C_{i-1}(w) \cup C_{i-2}(v)| \leq 2\alpha - 1$.

Montrons que $dd \geq \log(2\alpha - 1)$: on montre qu'il existe, avec grande probabilité, un motif dans l'arbre qui implique une dimension doublante minimale. Soit $u = u^{(i)}$ un sommet de T de niveau i de degré sortant α ayant comme propriétés :

1. $u^{(i-1)}$, la copie de u au niveau $i-1$, est de degré sortant α,
2. Soit $C = C_{i-1}(u^{(i)}) \cup C_{i-2}(u^{(i-1)})$. Tout arbre enraciné à un sommet $v \in C$ possède au moins deux feuilles.

Soit $\mathcal{E}_u$ l'évènement correspondant à l'existence d'un tel sommet u.

Montrons que si $\mathcal{E}_u$ est vérifié, alors $B_u(2^i)$ nécessite $(2\alpha - 1)$ boules de rayon 2^{i-1} pour être couverte.

Tout élément de $T \setminus T_u^{(i)}$ est à distance $\geq 2^{i+1}$. Aucun de ces éléments ne peut donc couvrir des points de $T_u^{(i)}$. Soit C_O un ensemble minimal de sommets de T qui soit une $2^{i-1}-$domination de $B_u(2^i)$.

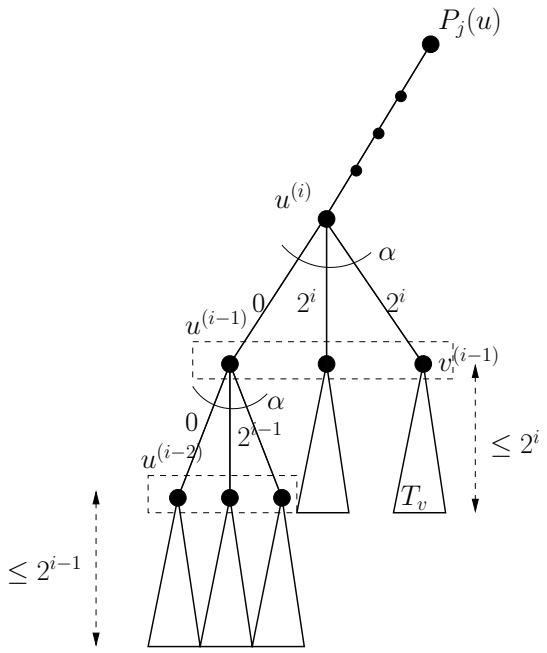


FIG. C.2: Configuration de sommets correspondant à l'évènement $\mathcal{E}_u$ dans T.

• Montrons que $\forall v \in C, T_v \cap C_O \neq \emptyset$ ce qui impliquerait que $|C_O| = |C|$ car $\forall v, v' \in C$ tq $\delta(v, v') > 0$, $T_v \cap T_{v'} = \emptyset$.

Soient $v \neq v' \in C$ tels que $\delta(v, v') > 0$. Par construction de T, $\delta(v, v') \geq 2^{i-1}$. Sachant que $|T_v| \geq 2$ et $|T_{v'}| \geq 2$, il existe $x \neq v \in T_v$ et $x' \neq v' \in T_{v'}$ tels que $\delta(x, v) > 0$ et $\delta(x', v') > 0$.

Ainsi, x (resp. x') ne peut être couvert par aucun élément de $T_{v'}$ (resp. T_v), donc $|C_O \cap T_v| \geq 1$.

Il existe $2\alpha - 1$ éléments de C situés à distance strictement positive les uns des autres. $B_u(2^i)$ nécessite donc $2\alpha - 1$ boules de rayon 2^{i-1} pour être couverte.

• Montrons maintenant que $\mathcal{E} = \bigcup_{S'=u \in S_{3k}, k \in [0, \frac{h}{3}]} \mathcal{E}_u$ est vérifié avec grande probabilité.

Soit $\mathcal{E}'_u$ l'évènement décrit par :

- $u = u^{(i)}$ est de degré sortant α,
- $u^{(i-1)}$ est de degré sortant α,
- Les sommets de $C_u^{i-2} \cup C_u^{i-1} \setminus u^{(i-1)} = C = \{v_1, v_2, .., v_{2\alpha-1}\}$ sont de degré sortant supérieur à 2.

Soit $\mathcal{E}' = \bigcup_{S'=u \in S_{3k}, k \in [0, \frac{h}{3}]} \mathcal{E}'_u$

$\mathbb{P}(\mathcal{E}'_u) \leq \mathbb{P}(\mathcal{E}_u)$ donc $\mathbb{P}(\mathcal{E}') \leq \mathbb{P}(\mathcal{E})$.

Soit F_X la variable aléatoire correspondant au nombre de fils du sommet X.

$$\mathbb{P}(\mathcal{E}') = \mathbb{P}(F_{u^{(i)}} = \alpha).\mathbb{P}(F_{u^{(i-1)}} = \alpha).\prod_{j=1}^{|C|}\mathbb{P}(F_{v_j} \geq 2) \quad (C.13)$$

$$= p_\alpha^2.(1 - \mathbb{P}(F_{v_j} \leq 1))^{2\alpha-2} \quad (C.14)$$

$$= p_\alpha^2.(1 - (p_0 + p_1))^{2\alpha-2} \quad (C.15)$$

Comme chaque sommet possède au plus α^2 descendants à deux niveaux, $|S'| \geq \frac{N}{\alpha^2}$. Pour tout $u \neq u' \in S'$, $\mathcal{E}'_u$ et $\mathcal{E}'_{u'}$ sont deux évènements indépendants donc $\mathbb{P}(\overline{\mathcal{E}'}) = \prod_{u \in S'} \mathbb{P}(\overline{\mathcal{E}'_u}) \leq (1 - \mathbb{P}(\mathcal{E}'_u))^{\frac{N}{\alpha^2}}$

Si $p_\alpha = \Omega(\sqrt{\frac{\log N}{N}})$ et $1 - (p_0 + p_1) = \theta(1)$ alors $\mathbb{P}(\overline{\mathcal{E}'}) = O(\frac{1}{N^c})$ pour c constant.

$\mathbb{P}(\mathcal{E}') = 1 - O(\frac{1}{N^c})$ et $\mathbb{P}(\mathcal{E}) = 1 - O(\frac{1}{N^c})$.

Donc avec grande probabilité, il existe un sommet u de T qui vérifie les propriétés 1,2,3. □

Dans la section 4.2, nous utilisons cette méthode pour produire des jeux de données de taille variable, de dimension doublante fixée et contenant des fragments. Ces jeux de données sont utilisés pour comparer les valeurs de stress produites par différents algorithmes de plongement d'un espace métrique en deux dimensions.

Dans le chapitre 4, nous avons voulu créer des jeux de données de dimension doublante donnée avec des fragments de taille fixée k. Nous définissons un fragment de la manière suivante :

Définition 69 (Fragment maximal)
Soit un espace métrique (S, δ) et t un seuil. Un fragment maximal est un sous-ensemble maximum de S tel que $\forall u, v \in S, \delta(u, v) \leq t$.

La Procédure `ajouter_fragments` de l'algorithme C.2 produit des fragments maximaux de taille k avec $t = 2$.

Conclusion

La méthode proposée permet de générer un grand nombre d'espaces métriques de dimension doublante donnée. Cependant, elle ne permet pas de générer "toutes" les métriques de dimension doublante donnée.

Nous avons fixé les distances entre sommets frères et ancêtres à l'aide de puissances de 2. En modifiant ce calcul des distances, on ne peut plus garantir la valeur de la dimension doublante de l'espace métrique généré.

Bibliographie

[1] A.A. Alatan, A.N. Akansu, and W. Wolf. Multi-modal dialog scene detection using hidden markov models for content-based multimedia indexing. *Multimedia Tools and Applications*, 14(2) :137–151, 2001.

[2] P. Androutsos, A. Kushki, K.N. Plataniotis, D. Androutsos, and A.N. Venetsanopoulos. Distributed MPEG-7 image indexing using small world user agents. In *Proc. of ICIP 2004*, pages 641–644, 2004.

[3] D. Archambault, T. Munzner, and D. Auber. Smashing peacocks further : Drawing quasi-trees from biconnected components. *Proc. of InfoVis 2006*, 12(5) :à paraître, September 2006.

[4] D. Arijon. *Grammar of the film language*. Silman-James Press, 1976.

[5] D. Auber. Using strahler numbers for real time visual exploration of huge graphs. In *Proc. of ICCVG 2002*, pages 56–69, 2002.

[6] D. Auber. *Graph Drawing Software*, chapter Tulip - A Huge Graph Visualization Framework. Verlag, 2003.

[7] D . Auber, Y. Chiricota, F. Jourdan, and G. Melancon. Multiscale visualization of small world networks. In *Proc. of INFOVIS '03*, pages 75–81, 2003.

[8] D. Auber, M. Delest, and Y. Chiricota. Strahler based graph clustering using convolution. In *Proc. of IV 2004*, pages 44–51, 2004.

[9] D. Auber, J.P. Domenger, S. Grivet, and G. Melançon. Bubble tree drawing algorithm. In *Proc. of ICCVG'04*, pages 633–641, 2004.

[10] H. Balakrishnan, M.F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Looking up data in p2p systems. *Commun. ACM*, 46(2) :43–48, 2003.

[11] G. Ball and D. Hall. Isodata : A novel method of data analysis and pattern classification. Technical Report AD699-616, Stanford Research Institute, Menlo Park, 1965.

[12] R.A. Becker and W.S. Cleveland. Brushing Scatterplots. *Technometrics*, 29 :127–142, 1987.

[13] B.B. Bederson and J.D. Hollan. Pad++ : a zooming graphical interface for exploring alternate interface physics. In *Proc. of ACM UIST '94*, pages 17–26. ACM Press, 1994.

[14] B.B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps : Making effective use of 2D space to display hierarchies. *ACM Trans. Graph.*, 21(4) :833–854, 2002.

[15] A.B. Benitez, M.J. Martinez, H. Rising, and P. Salembier. Color descriptors. In S. Manjunath et al. [16], chapter 13, pages 187–212.

[16] A.B. Benitez, M.J. Martinez, H. Rising, and P. Salembier. Description of a single multimedia document. In B. S. Manjunath, Phillipe Salembier, and Thomas Sikora, editors, *Introduction to MPEG-7 : Multimedia Content Description Language*, chapter 8, pages 111–127. Wiley, 2002.

[17] A.B. Benitez, M.J. Martinez, H. Rising, and P. Salembier. Motion descriptors. In S. Manjunath et al. [16], chapter 16, pages 261–279.

[18] J. Benois-Pineau, D. Barba, H. Nicolas, and A. Manoury. Domus videum, sous-projet structuration vidéo. Technical report, LaBRI, 2004.

[19] J. Benois-Pineau, W. Dupuy, and D. Barba. Recovering of visual scenarios in movies by motion analysis and grouping spatio-temporal colour signatures of video shots. In *Proc. of EUSFLAT'2001*, pages 385–390, September 5-7 2001 2001.

[20] C. Berge. *The Theory of Graphs.* Dover, 2001.

[21] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proc. of ICML '06 : Proceedings of the 23rd international conference on Machine learning*, pages 97–104. ACM Press, 2006.

[22] A. Del Bimbo. *Visual information retrieval.* Morgan Kaufmann Publishers Inc., 1999.

[23] B. Bollobas. *Modern Graph Theory.* Springer, 1998.

[24] C. Bonnet. *Manuel Pratique de Psychophysique.* Armand Colin, 1986.

[25] H. Le Borgne and N. O'Connor. Natural scene classification and retrieval using ridgelet-based image signatures. In *Proc. of Advanced Concepts for Intelligent Vision Systems, 2005*, 2005.

[26] N. Boujemaa, J. Fauqueur, M. Ferecatu, F. Fleuret, V. Gouet, B. Le Saux, and H. Sahbi. Ikona for interactive specific and generic image retrieval. In *Proc. of MMCBIR 2001*, 2001.

[27] S.K. Card, J.D. Mackinlay, and B.Shneiderman. Bifocal lens. pages 331–332, 1999.

[28] S.K. Card, G.G. Robertson, and W. York. The webbook and the web forager : An information workspace for the world-wide web. In *Proc. of CHI'96*, 1996.

[29] L. Carminati. *Détection et suivi d'objets dans les scènes de animées : application à la vidéo surveillance.* PhD thesis, Université Bordeaux 1, 2006.

[30] L. Carminati, J.Benois-Pineau, and M. Gelgon. Human detection and tracking for video surveillance applications in low density environment. *SPIE VCIP'2003*, 5150 :51–60, 2003.

[31] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Proc. of INFOVIS'1996*, pages 127–132, 1996.

[32] S-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong. VideoQ : an automated content based video search system using visual cues. In *Proc.of the fifth ACM Multimedia'97*, pages 313–324, 1997.

[33] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *J. Am. Stat. Assoc.*, 68 :361–368, 1973.

[34] L. Cieplinski, M. Kim, J. Ohm, M. Pickering, and A. Yamada. Text of ISO/IEC 15938-3/FCD Information technology - multimedia content description interface - Part 3 Visual, March 2001.

[35] Dorin Comaniciu and Peter Meer. Mean Shift : A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5) :603–619, May 2002.

[36] CIE :Commision Internationale de l'Eclairage. Internet address : http ://www.colour.org. 2006.

[37] Convera. ScreeningRoom. Internet address : http ://www.convera.com/, 2001.

[38] M. Cooper and J. Foote. Scene boundary via video self-similarity analysis. In *Proc. of IEEE ICIP'01*, pages 378–381, October 2001.

[39] C. Daassi. *Techniques d'interaction avec un espace de données temporelles.* PhD thesis, Université Joseph Fourier Grenoble 1, 2006.

[40] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4) :301–331, 1996.

[41] M. Delest, A. Don, and J. Benois-Pineau. Graph-based visual interfaces for navigation in indexed video contents. In *Proc. of third International Workshop on Content-Based Multimedia Indexing CBMI03*, pages 49–55. INRIA, Septembre 2003.

[42] M. Delest, A. Don, and J. Benois-Pineau. Intuitive color-based visualization of multimedia content as large graphs. In *Proc. of Electronic Imaging, Visualization and Data Analysis 2004, EI04*, volume 5295, pages 65–74. SPIE, Janvier 2004.

[43] M. Delest, A. Don, and J. Benois-Pineau. DAG-based visual interfaces for navigation in indexed video content. *Multimedia Tools and Applications*, 31(1) :51–72, Octobre 2006.

[44] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *em* algorithm. *Journal of the Royal Statistical Society*, 39(1) :1–38, 1977.

[45] A. Don and N. Hanusse. A deterministic multidimensional scaling algorithm for data visualisation. In *Proc. of 10th Int. Conf. on Information Visualisation, IV06*, pages 511–520. IEEE, Juin 2006.

[46] A. Don and N. Hanusse. Searching in collections of indexed images using greedy routing. Technical Report RR-1419-06, LaBRI, 2006.

[47] A. Don, L.Carminati, and J. Benois-Pineau. Detection of visual dialog scenes in video content based on structural and semantic features. In *Proc. of fourth International Workshop on Content-Based Multimedia Indexing CBMI05*, pages 1–8. Tampere University of Technology, Juin 2005.

[48] M. Durik and J. Benois-Pineau. Robust motion characterisation for video indexing based on MPEG-2 optical flow. In *Proc. of CBMI 2001*, pages 57–64, 2001.

[49] T. Dwyer and Y. Koren. Dig-cola : Directed graph layout through constrained energy minimization. In *Proc. of INFOVIS '05*, page 9, Washington, DC, USA, 2005. IEEE Computer Society.

[50] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42 :149–160, 1984.

[51] P. Eades. Drawing free trees. *Bulletin of the institute for combinatorics and its applications*, 5 :10–36, 1992.

[52] A. P. Ershov. On programming of arithmetic operations. *Com. of the A.C.M*, 8 :3–6, 1958.

[53] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content : The QBIC system. *IEEE Computer*, 28(9) :23–32, 1995.

[54] Flickr. Flickr. Internet address : http ://www.flickr.com/, 2006.

[55] P-M. Fonseca and J. Nesvadba. Face tracking in the compressed domain. *EURASIP Journal on Applied Signal Processing*, 2006 :Article ID 59451, 11 pages, 2006.

[56] E. Freeman and D. Gelernter. Lifestreams : A storage model for personal data. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 25(1) :80–86, 1996.

[57] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proc. DIMACS Int. Work. Graph Drawing, GD*, pages 388–403. Springer-Verlag, 1994.

[58] T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11) :1129–1164, 1991.

[59] G. W. Furnas. Generalized fisheye views. In *Proc. of CHI '86*, pages 16–23, New York, NY, USA, 1986. ACM Press.

[60] P. Gajer and S.G. Kobourov. GRIP : Graph dRawing with intelligent placement. In *Graph Drawing*, pages 222–228, 2000.

[61] J. Gao, L.J. Guibas, and A. Nguyen. Deformable spanners and applications. In *Proc. of SCG '04*, pages 190–199. ACM Press, 2004.

[62] Google. Google. Internet address : http ://www.google.com/, 2006.

[63] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. In *Proc. of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 326–335. ACM Press, 2002.

[64] R.M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1 :4–29, April 1984.

[65] D.M. Green and J.A. Swets. *Signal Detection Theory and Psychophysics*. John Wiley&Sons, 1974.

[66] V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *Computer*, 28(9) :18–22, 1995.

[67] B. Gunsel, A. Ferman, and A. Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking. In *Journal of Electronic Imaging*, volume 7(3), pages 592–604, 1998.

[68] A. Guttman. R-trees : a dynamic index structure for spatial searching. In *SIGMOD '84 : Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57. ACM Press, 1984.

[69] S. Hachul and M. Jünger. Large-graph layout with the fast multipole multilevel method. Technical Report 68W01, Zentrum für Angewandte Informatik Köln, Lehrstuhl Jünger, December 2005.

[70] S. Hachul and M. Jünger. An experimental comparison of fast algorithms for drawing general large graphs. In Patrick Healy and Nikola S. Nikolov, editors, *Graph Drawing*, pages 235–250. Springer, 2006.

[71] A. Hanjalic. Shot-boundary detection : unraveled and resolved ? *IEEE Trans. Circuits and Systems for Video Technology*, 12(2) :90–105, February 2002.

[72] S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. In *Proc. of SCG '05 : Proceedings of the twenty-first annual symposium on Computational geometry*, pages 150–158. ACM Press, 2005.

[73] D. Harel and Y. Koren. A fast multi-scale method for drawing large graphs. *Journal of Graph Algorithms and Applications*, 3 :179–202, 2002.

[74] T. Harris. *The theory of branching processes*. Springer, 1963.

[75] A. Hauptmann and M. Smith. Text, speech, and vision for video segmentation : The Informedia project. In *Proc. of AAAI Fall 1995 Symposium on Computational Models for Integrating Language and Vision*, 1995.

[76] M.A. Hearst. Clustering versus faceted categories for information exploration. *Communication of the ACM*, 49(4) :59–61, 2006.

[77] R. Hjelsvold, S. Landorgen, R. Midtstraum, and O. Sandstaa. Integrated Video Archive Tools. In *Proc. of the ACM Multimedia'95*, pages 283–293, San Francisco, California, 1995.

[78] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[79] E. Huang. Exploiting the intrinsic dimensionality of data for nearest neighbor search. Master's thesis, Princeton, 2005.

[80] B. Huet, N.Kern, G. Guarascio, and B. Merialdo. Relational skeletons for retrieval in patent drawings. In *Proc. of ICIP'01*, volume 2, pages 737–740, 2001.

[81] IBM. QBIC at hermitage museum. Internet address : http ://www.hermitagemuseum.org/, 2001.

[82] InfoVis. InfoVis. Internet address : http ://www.infovis.org/infovis/, 2006.

[83] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1 :69–91, 1985.

[84] B. Johnson and B. Shneiderman. Tree-maps : a space-filling approach to the visualization of hierarchical information structures. In *Proc. of VIS '91*, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[85] P. Joly and K-K. Kim. *Efficient Automatic Analysis of Camera Work and Microsegmentation of Video Using Spatio-Temporal Images*, volume 8(4) of *Signal Processing : Image Communication*, pages 295–307. Elsevier, Eurasip, Amsterdam, mai 1996.

[86] F. Jourdan and G. Melançon. Multiscale hybrid mds. In *Proc. of IV'2004*, pages 388–393, 2004.

[87] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1) :7–15, 1989.

[88] Y-L. Kang, J-H. Lim, M.S. Kankanhalli, C. Xu, and Q. Tian. Goal detection in soccer video using audio/visual keywords. In *Proc. of ICIP 2004*, pages 1629–1632, 2004.

[89] D. R. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proc. of STOC '02*, pages 741–750. ACM Press, 2002.

[90] D.A. Keim. Designing pixel-oriented visualization techniques : Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1) :59–78, 2000.

[91] K. Knoblauch, C. Charrier, H. Cherifi, J.N. Yang, and L.T. Maloney. Difference scaling of image quality in compression-degraded images. *Perception*, 27 :174–, 1998.

[92] R. Krauthgamer and J.R Lee. The intrinsic dimensionality of graphs. In *Proc. of STOC 2003*, pages 438–447, 2003.

[93] R. Krauthgamer and J.R. Lee. Navigating nets : simple algorithms for proximity search. In *Proc. of SODA '04 : Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 798–807, 2004.

[94] J. Kreyß, M. Röper, P. Alshuth, T. Hermes, and O. Herzog. Video retrieval by still-image analysis with imageminer. In *Proc. of SPIE Electronic Imaging 1997*, pages 36–44, 1997.

[95] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29 :1–27, 1964.

[96] J.B. Kruskal and M.Wish. *Multidimensional Scaling (Quantitative Applications in the Social Sciences)*. SAGE Publications, 1978.

[97] J. Lamping and R. Rao. Laying out and visualizing large trees using a hyperbolic space. In *Proc. of ACM UIST '94*, pages 13–14, New York, NY, USA, 1994. ACM Press.

[98] C. Langreiter. Retrievr. Internet address : http ://labs.systemone.at/retrievr/, 2006.

[99] J. L.Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9) :509–517, 1975.

[100] R. Leonardi, P. Migliorati, Adami, A. Bugatti, and L. Rossi. Describing multimedia documents in natural and semantic-driven ordered hierarchies. In *Proc. of ICASSP 2000*, pages 2023–2027, July 2000.

[101] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110, 2004.

[102] J. Luo and A.E. Savakis. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. In *Proc. of ICIP'01*, volume 2, pages 745–748, 2001.

[103] J.D. Mackinlay, G.G. Robertson, and S.K. Card. The perspective wall : detail and context smoothly integrated. In *Proc. of CHI '91*, pages 173–176, New York, NY, USA, 1991. ACM Press.

[104] J. MacQueen. Some methods for classifications and analysis of multivariate observations. In *In Proc. Fifth Berkeley Symp. Math. Stat. and Prob.*, pages 281–297. University of California Press, 1967.

[105] L.T. Maloney and J.N. Yang. Maximum likelihood difference scaling. *Journal of Vision*, 3(8) :573–585, 10 2003.

[106] F. Manerba, R. Leonardi, and J. Benois-Pineau. Real-time extraction of foreground objects in a rough indexing framework. In *Proc. of WIAMIS'2005*, Montreux, SW, 2005.

[107] C. Mar-Molinero and C. Serrano-Cinca. Bank failure : a multidimensional scaling approach. *European Journal of Finance*, 7(2) :165–183, June 2001.

[108] J.F. Marckert. Arbres et chemins. In *Proc. of ALEA 2003*, pages 1–39, 2003.

[109] J. M. Martinez. MPEG-7 overview (version 10), iso/iec jtc1/sc29/wg11n6828, October 2004.

[110] T. Meiers, T. Sikora, and I. Keller. Hierarchical image database browsing environment with embedded relevance feedback. In *Proc. of ICIP*, volume 2, pages 593–596, 2002.

[111] Microsoft. World Wideo Media Exchange. Internet address : http ://wwmx.org/, 2005.

[112] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. of ECCV '02*, pages 128–142. Springer-Verlag, 2002.

[113] P.A. Mlsna and N.M. Sirakov. Intelligent shape feature extraction and indexing for efficient content-based medical image retrieval. In *Proc. of SouthWest'04*, pages 172–176, 2004.

[114] A. Mojsilovic and B. Rogowitz. Capturing image semantics with low-level descriptors. In *Proc. of ICIP'01*, volume 1, pages 18–21, 2001.

[115] A. Mojsilovic and B. Rogowitz. Semantic metric for image library exploration. *IEEE Transactions on Multimedia*, 6 :828–838, 2004.

[116] J.W. Moon and L. Moser. Some packing and covering theorems. *Colloquium Mathematicum*, 17 :103–110, 1967.

[117] R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. In *Proc. of ACM SIGCOMM 2001*, pages 149–160, San Diego, CA, September 2001.

[118] A. Morrison and M. Chalmers. A pivot-based routine for improved parent-finding in hybrid mds. *Proc. of INFOVIS'2003*, 3(2) :109–122, 2004.

[119] A. Morrison, G. Ross, and M. Chalmers. Fast multidimensional scaling through sampling, springs and interpolation. In *Proc. of INFOVIS'2003*, volume 2, pages 68–77, 2003.

[120] R. Motwani and P. Raghavan. *Randomized algorithms*, chapter Chapter 4 : Tail inequalities. Cambridge University Press, 1995.

[121] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, 2004.

[122] H. Nicolas, A. Manoury, J. Benois-Pineau, W. Dupuis, and D. Barba. Grouping video shots into scenes based on 1D mosaic descriptors. In *Proc. of ICIP 2004*, volume 1, pages 637–640, 2004.

[123] NIST. Trec video retrieval evaluation. Internet address : http ://www-nlpir.nist.gov/projects/trecvid/, 2005.

[124] G. Obein, K. Knoblauch, and F. Viénot. Difference scaling of gloss : Nonlinearity, binocularity, and constancy. *J. Vis.*, 4(9) :711–720, 8 2004.

[125] J-M. Odobez, D. Gatica-Perez, and Mael Guillemot. Video shot clustering using spectral methods. In INRIA IRISA, editor, *Proc. of CBMI'03*, pages 95–101, 2003.

[126] E. Osuna, R. Freund, and F. Girosi. Training Support Vector Machines : an application to face detection. In *Proc. of IEEE CVPR 1997*, pages 130–136, 1997.

[127] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6) :559–572, 1901.

[128] S. Pfeiffer, R. Lienhart, and W. Effelsberg. Scene determination based on video and audio features. In *Proc. of the CMCS'99*, pages 685–690, 1999.

[129] Photomesa. Photomesa. Internet address : http ://www.photomesa.com/, 2005.

[130] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines : visualizing personal histories. In *Proc. of CHI '96*, New York, NY, USA.

[131] S. Pook, E. Lecolinet, G. Vaysseix, and E. Barillot. Context and interaction in zoomable user interfaces. In *Proc. of AVI '00*, pages 227–231, 2000.

[132] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[133] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable and content-adressable network. In *ACM SIGCOMM 2001*, pages 161–172, San Diego, CA, September 2001.

[134] E. M. Reingold and J. S. Tilford. Tidier drawing of trees. *IEEE Transactions on Software Engineering*, 7(2) :223–228, 1981.

[135] J. Rekimoto and M. Green. The information cube : Using transparency in 3D information visualization. In *Proc. of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, pages 125–132, 1993.

[136] G.G. Robertson, J.D. Mackinlay, and S.K. Card. Cone trees : animated 3D visualizations of hierarchical information. In *Proc. of CHI '91*, pages 189–194. ACM Press, 1991.

[137] H. A. Rowley, S. Baluja, and T. Kanade. Human face recognition in visual scene. Technical Report CMU-CS-95-158R, Carnegie Mellon University, 1995.

[138] C. Saraceno and R. Leonardi. Audio as a support to scene change detection and characterization of video sequences. In *Proc. of ICASSP '97*, pages 0–4, 1997.

[139] C. Saraceno and R. Leonardi. Identification of story units in audio-visual sequencies by joint audio and video processing. In *Proc. of ICIP'98*, pages 363–368, October 1998.

[140] M. Seo, B. Ko, H. Chung, and J. Nam. ROI-based medical image retrieval using human-perception and MPEG-7 visual descriptors. In *Proc. of CIVR'06*, pages 231–240, 2006.

[141] J. Serra. *Image Analysis and Mathematical Morphology.* Ac. Press, London, 1982.

[142] B. Shneiderman. Tree visualization with tree-maps : 2D space-filling approach. *ACM Trans. Graph.*, 11(1) :92–99, 1992.

[143] B. Shneiderman. *Designing the user interface : strategies for effective human-computer interaction.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, third edition edition, 1998.

[144] SmartMoney. Smartmoney. Internet address : http ://www.smartmoney.com/marketmap/, 2006.

[145] J.R. Smith and S-F. Chang. An image and video search engine for the world-wide web. In *Proc. of SPIE Electronic Imaging*, volume 5, San Jose, CA, February 1997.

[146] A. N. Strahler. Hypsomic analysis of erosional topography. *Bulletin Geological Society of America*, 63 :1117–1142, 1952.

[147] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understandings of hierarchical system structures. *IEEE Transactions in Systems, Man, and Cybernetics*, 11(2) :109–125, 1981.

[148] H. Sundaram and S-F Chang. Computable scenes and structures in films. *IEEE Transactions on Multimedia*, 4(4) :482–491, 2002.

[149] S. Tamura. Clustering based on multiple paths. *Pattern Recognition*, 15(6) :477–483, 1982.

[150] W. Tavanapong and J. Zhou. Shot clustering techniques for story browsing. *IEEE Transactions on Multimedia*, 6(4) :517–527, August 2004.

[151] J. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framwork for nonlinear dimensionality reduction. *Science*, (290) :2319–2323, 2000.

[152] J.B. Tenenbaum, V. De Sialva, and J.C. Langford. Isomap homepage. Internet address : http ://isomap.stanford.edu/, (last visited 20/03/2005), 2005.

[153] M. Thorup and U. Zwick. Approximate distance oracles. *Journal of the ACM*, 52(1) :1–24, 2005.

[154] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. VideoMAP and VideoSpaceIcon : tools for anatomizing video content. In *Proc. of CHI '93*, pages 131–136, 1993.

[155] W.S. Torgerson. Multidimensional scaling of similarity. *Psychometrika*, 30(4) :379–393, December 1965.

[156] E.R. Tufte. *The visual display of quantitative information.* Graphics Press, 1983.

[157] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE CVPR 1991*, pages 71–86, 1991.

[158] S. Uchihashi, J. Foote, A.Girgensohn, and J. Boreczky. Video Manga : Generating semantically meaningful video summaries. In ACM Press, editor, *Proc. of ACM Multimedia 1999*, pages 383–392, 1999.

[159] A. Vailaya, H.J. Zhang, C. Yang, F.I. Liu, and A.K. Jain. Automatic image orientation detection. In *ICIP99*, volume 2, pages 600–604, 1999.

[160] J.J. van Wijk and E.R. van Selow. Cluster and calendar based visualization of time series data. In *Proc. of INFOVIS'99*, pages 4–9, 1999.

[161] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York,, 1995.

[162] Virage. VideoLogger. Internet address : http ://www.virage.com/, 2001.

[163] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. of CHI '04*, pages 319–326, New York, NY, USA, 2004. ACM Press.

[164] W. W. Cleveland and R. McGill. Graphical perception : theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387) :531–554, 1984.

[165] J. Q. Walker. A node-positioning algorithm for general trees. *Softw. Pract. Exper.*, 20(7) :685–705, 1990.

[166] A.B. Watson and L. Kreslake. Measurement of visual impairment scales for digital video. In *Proceedings of the SPIE EI'01*, volume 4299, pages 79–89, 2001.

[167] H. W. Watson and F. Galton. On the probability of extinction of families. *Journal of the Anthropological Institute of Great Britain and Ireland*, 4 :138–144, 1875.

[168] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 :440–442, June 1998.

[169] M. Weber, M. Alexa, and W. Muller. Visualizing time-series on spirals. In *Proc. of INFOVIS '01*, pages 7–14, 2001.

[170] J.J. Van Wijk and H. van de Wetering. Cushion treemaps : Visualization of hierarchical information. In *Proc. of INFOVIS '99*, pages 73–78, Washington, DC, USA, 1999. IEEE Computer Society.

[171] A. Aner Wolf and J.R. Kender. Video summaries and cross-referencing through mosaic-based representation. *CVIU*, 95(2) :201–237, August 2004.

[172] J.K Wu, C.P. Lam, B.M. Mehtre, Y.J. Gao, and A.D. Narasimhalu. Content-based retrieval for trademark registration. *Multimedia Tools and Applications*, 3(3) :245–267, 1996.

[173] W. Xiong, B. Qiu, Q. Tian, C. Xu, S.H. Ong, and K. Foong. Content-based medical image retrieval using dynamically optimized regional features. In *Proc. of ICIP'05*, volume 3, pages III : 1232–1235, 2005.

[174] M-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images : a survey. *IEEE PAMI*, 24(1) :34–58, January 2002.

[175] A.C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. Technical Report CS-TR-77-642, Stanford, CA, USA, 1977.

[176] M. Yeung and B.L. Yeo. Time-constrained clustering for segmentation of video into story units. In *Proc. of ICPR '96*, pages 375–380, August 1996.

Indexation et navigation dans les contenus visuels : approches basées sur les graphes

Résumé : Cette thèse s'inscrit dans le domaine de l'indexation et de la visualisation des documents vidéo et des collections d'images. Les méthodes proposées reposent sur l'utilisation de graphes pour représenter les relations de similarité entre les plans vidéo et images indexés.

La première partie de cette thèse concerne l'indexation des documents vidéo en scènes. Les scènes sont des ensembles de plans vidéo partageant des caractéristiques similaires. Nous proposons d'abord une méthode interactive de détection de groupes de plans, partageant un contenu couleur similaire, basé sur la fragmentation de graphe. Nous abordons ensuite l'indexation des documents vidéo en scènes de dialogue, basée sur des caractéristiques sémantiques et structurelles présentes dans l'enchaînement des plans vidéo.

La seconde partie de cette thèse traite de la visualisation et de la recherche dans des collections d'images indexées. Nous présentons un algorithme de plongement d'un espace métrique dans le plan appliqué à la visualisation de collections d'images indexées. Ce type de visualisation permet de représenter les relations de dissimilarité entre images et d'identifier visuellement des groupes d'images similaires. Nous proposons enfin une interface de recherche d'images basée sur le routage local dans un graphe. Les résultats d'une validation expérimentale sont présentés et discutés.

Mots-clef : Graphes, indexation multimédia, analyse vidéo, visualisation d'information, interaction.
Discipline : Informatique

Searching and indexing visual contents : graph-based approaches

Abstract : This thesis deals with the indexation and the visualisation of video documents and collections of images. The proposed methods are based on graphs to represent similarity relationships between indexed video shots and images.

The first part of this thesis deals with the indexation of video documents into scenes. A scene is a set of video shots that share common features. We first propose an interactive method to group shots with similar color content using graph clustering. We then present a technique to index video documents into dialogue scenes based on semantic and structural features.

The second part of this thesis deals with visualisation and search in collections of indexed images. We present an algorithm for embedding a metric space in the plane applied to collections of indexed images. The aim of this technique is to visualise the dissimilarity relationships between images to identify clusters of similar images. Finally, we present a user interface for searching images, inspired from greedy routing in networks. Results from experimental validation are presented and discussed.

Keywords : Graphs, multimedia indexing, video analysis, information visualisation, interaction.
Field : Computer Science