Passage à l'échelle pour la visualisation interactive exploratoire de données : approches par abstraction et par déformation spatiale

Gaëlle Richer

LABORATOIRE BORDELAIS DE RECHERCHE EN INFORMATIQUE

LaBRI

université de BORDEAUX

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **Gaëlle Richer**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Passage à l'échelle pour la visualisation interactive exploratoire de données : approches par abstraction et par déformation spatiale

---

Date de soutenance : 26 novembre 2019

Sous la direction de : David Auber et Romain Bourqui

Devant la commission d'examen composée de :

Caroline Appert	Directrice de Recherche, CNRS	Rapportrice
Florence Sèdes	Professeur, Université Paul Sabatier, Toulouse III	Rapportrice
Guy Melançon	Professeur, Université de Bordeaux	Président du jury
Arnaud Sallaberry	Maître de Conférences, Université Paul Valéry, Montpellier III	Examinateur
David Auber	Professeur, Université de Bordeaux	Directeur
Romain Bourqui	Maître de Conférences, Université de Bordeaux	Co-directeur

– 2019 –

LABORATOIRE D'ACCUEIL : LaBRI (UMR 5800), 351 cours de la Libération, 33405 Talence, France

DURÉE : septembre 2016 – novembre 2019

TITRE : Passage à l'échelle pour la visualisation interactive exploratoire de données : approches par abstraction et par déformation spatiale

RÉSUMÉ : La visualisation interactive est un outil essentiel pour l'exploration, la compréhension et l'analyse de données. L'exploration interactive efficace de jeux de données grands ou complexes présente cependant deux difficultés fondamentales. La première est visuelle et concerne les limitations de la perception et cognition humaine, ainsi que celles des écrans. La seconde est computationnelle et concerne les limitations de capacité mémoire ou de traitement des machines standards. Dans cette thèse, nous nous intéressons aux techniques de passage à l'échelle relativement à ces deux difficultés, pour plusieurs contextes d'application.

Pour le passage à l'échelle visuelle, nous présentons une approche versatile de mise en évidence de sous-ensembles d'éléments par déformation spatiale appliquée aux vues multiples et une représentation abstraite et multi-échelle de coordonnées parallèles. Sur les vues multiples, la déformation spatiale vise à remédier à la diminution de l'efficacité de la surbrillance lorsque les éléments graphiques sont de taille réduite. Sur les coordonnées parallèles, l'abstraction multi-échelle consiste à simplifier la représentation tout en permettant d'accéder interactivement au détail des données, en les pré-agrégeant à plusieurs niveaux de détail.

Pour le passage à l'échelle computationnelle, nous étudions des approches de pré-calcul et de calcul à la volée sur des infrastructures distribuées permettant l'exploration de jeux de données de plus d'un milliard d'éléments en temps interactif. Nous présentons un système pour l'exploration de données multi-dimensionnelles dont les interactions et l'abstraction respectent un budget en nombre d'éléments graphiques qui, en retour, fournit une borne théorique sur les latences d'interactions dues au transfert réseau entre client et serveur. Avec le même objectif, nous comparons des stratégies de réduction de données géométrique pour la reconstruction de cartes de densité d'ensembles de points.

MOTS-CLÉS : visualisation, passage à l'échelle, interface homme-machine, données massives, déformation spatiale, abstraction de données

---

TITLE : Addressing scaling challenges in interactive exploratory visualization with abstraction and spatial distortion

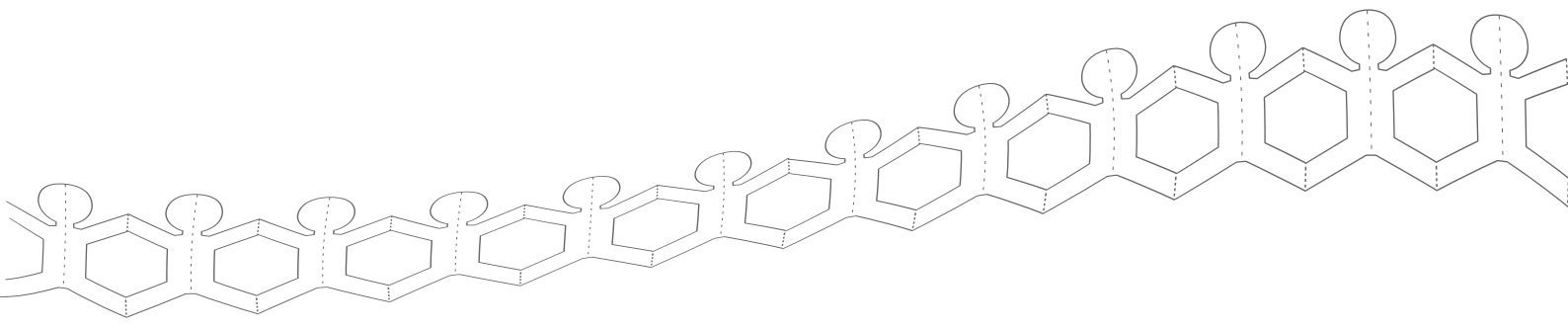
ABSTRACT : Interactive visualization is helpful for exploring, understanding, and analyzing data. However, increasingly large and complex data challenges the efficiency of visualization systems, both visually and computationally. The visual challenge stems from human perceptual and cognitive limitations as well as screen space limitations while the computational challenge stems from the processing and memory limitations of standard computers. In this thesis, we present techniques addressing the two scalability issues for several interactive visualization applications.

To address visual scalability requirements, we present a versatile spatial-distortion approach for linked emphasis on multiple views and an abstract and multi-scale representation based on parallel coordinates. Spatial distortion aims at alleviating the weakened emphasis effect of highlighting when applied to small-sized visual elements. Multiscale abstraction simplifies the representation while providing detail on demand by pre-aggregating data at several levels of detail.

To address computational scalability requirements and scale data processing to billions of items in interactive times, we use pre-computation and real-time computation on a remote distributed infrastructure. We present a system for multi-dimensional data exploration in which the interactions and abstract representation comply with a visual item budget and in return provides a guarantee on network-related interaction latencies. With the same goal, we compared several geometric reduction strategies for the reconstruction of density maps of large-scale point sets.

KEYWORDS : information visualization, human-computer interface, scalability, large-scale data, spatial distortion, data abstraction

À mon grand-père.
1936 – 2018



Remerciements

Je tiens à remercier tout d'abord Caroline Appert et Florence Sèdes pour avoir accepté de rapporter mon manuscrit, et Guy Melançon et Arnaud Sallaberry d'avoir accepté de participer à mon jury. Un grand merci à Caroline Appert et Arnaud Sallaberry pour leurs corrections minutieuses du manuscrit. J'ai été encadré pendant quatre ans par David Auber et Romain Bourqui que je remercie vivement pour leur énergie et soutien, leurs conseils et critiques, leur temps et patience, leurs discussions et digressions et pour avoir lu *tous les mots*. Je remercie aussi Xavier Blanc et Nicolas Bonichon d'avoir suivi le déroulement du doctorat et prodigué de bons conseils. Merci aussi, pour leur soutien et sympathie, à l'ensemble des membres et ex-membres du groupe EVADoMe/B^3, au personnel du laboratoire et aux enseignants de l'IUT que j'ai pu rencontrer. Merci enfin à mes anciens enseignants pour leurs encouragements sporadiques, au détour d'un couloir.

Un grand merci à Frédéric et Aarón, précieux compagnons de fortune et d'infortune, ainsi qu'Alexandre, Jason, Norbert, Myriam et tous les autres avec qui j'ai partagé discussions, jeux, cafés ou foulées. David est à remercier pour avoir créé l'étincelle initiatrice de plus de *1000 km* parcourus à baskets durant ces quatre années. Merci également à Marthe pour la nourriture, Vincent pour les jeux et Rémi pour les aventures.

Une pensée enfin à la fidèle équipe des jeux de société nocturnes (Aurélien, Benoît, Simon, Nicolas, etc) et spécialement à Gaëtan qui a finalement relu quelques pages de ce manuscrit.

> But life is short and information endless : nobody has time for everything. Abbreviation is a necessary evil and the abbreviator's business is to make the best of a job which, though intrinsically bad, is still better than nothing. He must learn to simplify, but not to the point of falsification.
>
> — Aldous Huxley, *Brave New World*

Table des matières

1 Introduction

L'évolution des composants électroniques (processeurs, mémoire) et le développement des réseaux informatiques a conduit à une augmentation exponentielle de la quantité de données générées et stockées au cours des quarante dernières années [80]. En 2013, on estimait à *1200* exaoctets la quantité de données stockées [112] et à plus de 10 téraoctets la quantité de données générées chaque seconde en 2015 [110]. Cette tendance se reflète directement sur l'évolution de l'espace de stockage de dépôts de données tel que celui de l'institut européen de bio-informatique EMBL-EBI (*20* pétaoctets en 2012 et presque *80* pétaoctets en 2015 [41]), mais aussi sur la taille des jeux de données étudiés. En astronomie par exemple, il est prévu que le télescope LSST [1] collecte durant dix ans environ *500* pétaoctets pour former un jeu de données final de *15* pétaoctets [83]. Le web est un exemple de réseau immense et grandissant qu'on estime à plus de *300* milliards de nœuds (pages) aujourd'hui [2]. Cette croissance est un atout car l'analyse de ces données permet de prendre des décisions plus rapidement, par exemple dans le cas de données cliniques, et peut également mener à de nouvelles découvertes scientifiques, par exemple dans le cas de données de simulation large-échelle en physique.

Par ailleurs, l'avènement de l'informatique et la démocratisation des ordinateurs à interface graphique dans les années 50 a permis la création du domaine de l'informatique graphique et ainsi de la visualisation d'information. Cette-dernière consiste en la représentation graphique *automatique* de données et succède aux représentations graphiques dessinées à la main utilisées depuis le XVIII[e] siècle [158] comme le diagramme de Minard de la figure 1.2 représentant le volume de marchandises transportées sur un canal entre chacune de ses sections en 1862. La visualisation est définie par Card et al. [26] comme « l'utilisation de représentations visuelles des données assistée par ordinateur pour amplifier la cognition ». Cette amplification concerne plusieurs aspects des processus cognitifs, par exemple la mémoire, l'identification de motifs, ou la recherche d'information [59]. L'efficacité de la visualisation d'information pour analyser des données provient des capacités de perceptions visuelles humaines, ainsi les choix d'abstraction visuelle utilisée pour encoder les données (p. ex. la couleur ou la taille) reposent sur des théories de neuropsychologie comme les lois *Gestalt* pour la perception de motifs et le traitement *pré-attentif* [155] pour l'effet de saillie.

L'interactivité est une part importante de la visualisation qui implique l'utilisateur dans le processus de génération de l'image et lui permet l'*exploration* des données par altérations successives de la représentation visuelle. Ce mécanisme est notamment indispensable lorsque l'objectif de l'outil de visualisation n'est pas de présenter, informer ou observer des données mais de les explorer pour les analyser et les comprendre. L'interaction existait déjà pour les formats physiques de visualisation utilisés au XX[e] siècle, comme l'illustre les matrices ré-ordonnables de Bertin de la figure 1.1. Chaque ligne de la matrice est dessinée sur une fiche de sorte que les lignes puissent être ré-ordonnées aisément. L'interactivité devient particulièrement essentielle pour l'analyse des données lorsque la taille de celles-ci grandit puisque une vue unique ne peut plus permettre de représenter l'ensemble de l'information des données.

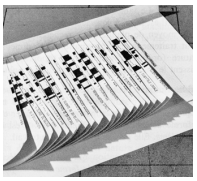


Figure 1.1: Matrices matériellement ré-ordonnables de Bertin [14] utilisant une fiche par ligne, ce qui permet de reclasser manuellement les lignes par ré-agencement des fiches.

1.1 Problématique

Un outil d'exploration visuel *efficace* pour un jeu de données peut perdre en efficacité dès lors qu'il est utilisé pour un autre de plus grande taille, par détérioration de l'utilisabilité de sa représentation visuelle ou de la réactivité de ses interactions. Le facteur majeur diminuant l'efficacité d'une représentation est l'encombrement visuel et, en particulier, la superposition des éléments

1. Large Synoptic Survey Telescope
2. Selon les archives de l'Internet Archive : *357* milliards de pages au dernier accès

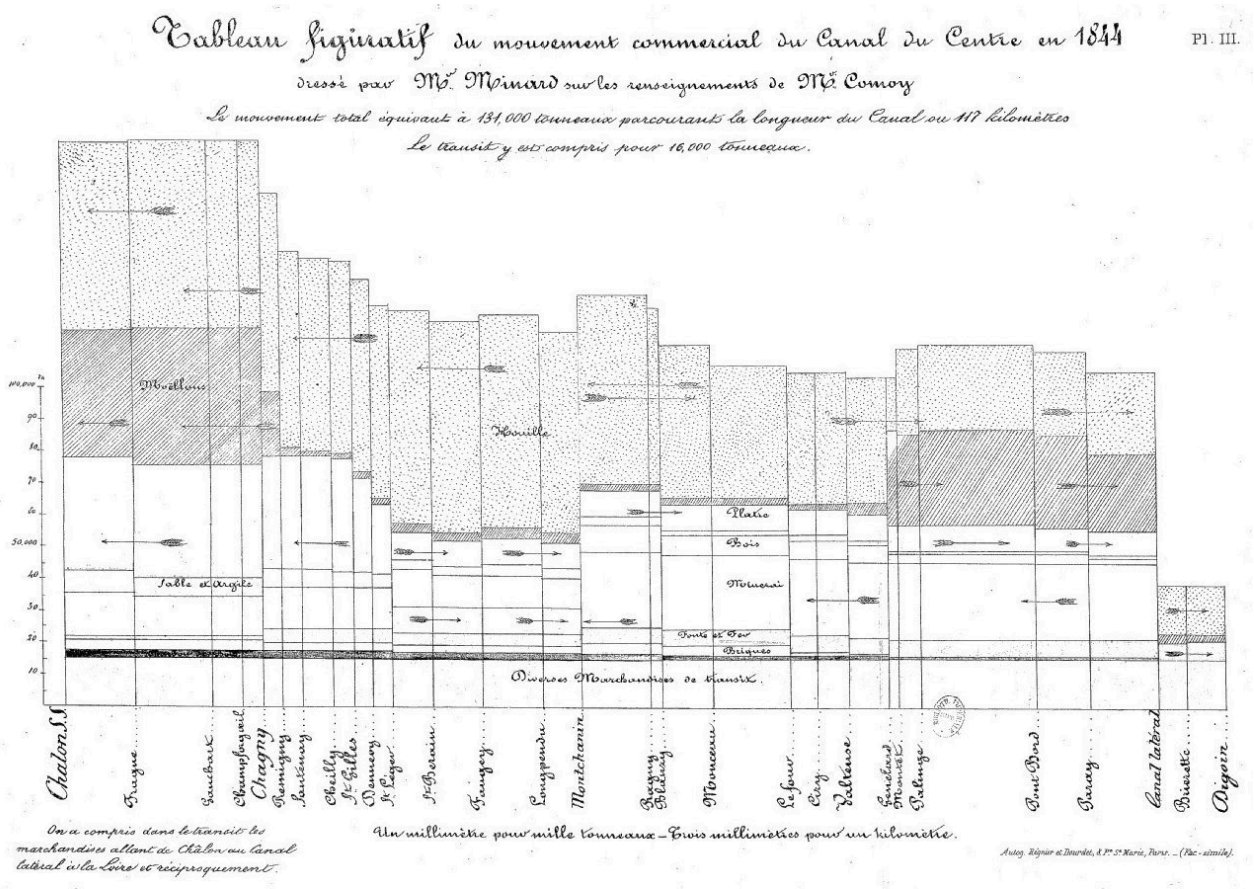


FIGURE 1.2: Tableau graphique de Charles-Joseph Minard représentant le transport de marchandises sur le canal du Centre. La hauteur de chaque barre correspond à la charge (en nombre de tonneaux) et sa largeur à la distance de l'étape (en kilomètres), de sorte que l'aire représente le coût de transport (source : BnF [116]).

graphiques qui cachent de l'information. La diminution de la réactivité des interactions devient néfaste à partir d'un seuil de latence qui dépend notamment du type de tâches accomplies par l'utilisateur. En effet, des latences additionnelles de *500 ms* peuvent significativement diminuer la quantité d'observations notée par l'utilisateur et affecter négativement l'étendue de leur analyse [106]. Bien que ce soit, en partie, les avancées techniques du matériel informatique qui sont à l'origine de la croissance exponentielle de la taille des données collectées et analysées, ces mêmes avancées ne sont pas toujours suffisantes pour conjointement permettre le traitement de ces données en temps interactif [4, 66].

Les jeux de données considérés *grands* pour la visualisation comportent de plusieurs millions à plus d'un milliard d'éléments [107] ou plus d'un téraoctet de données [66] et constituent un défi pour les systèmes de visualisation conventionnels qui font face à deux verrous. Le premier verrou est visuel : le paradigme *atomique*, qui représente un élément graphique pour chaque élément des données, devient inutilisable lorsque le nombre d'entités des données dépasse la définition des écrans. Le second verrou est computationnel : les machines standards actuelles ne disposent pas de ressources suffisantes pour permettre le traitement de tels jeux de données en temps *interactif*. Pour permettre l'exploration de tels jeux de données, les systèmes de visualisation interactive doivent donc posséder des mécanismes de *passage à l'échelle* relativement à ces deux verrous.

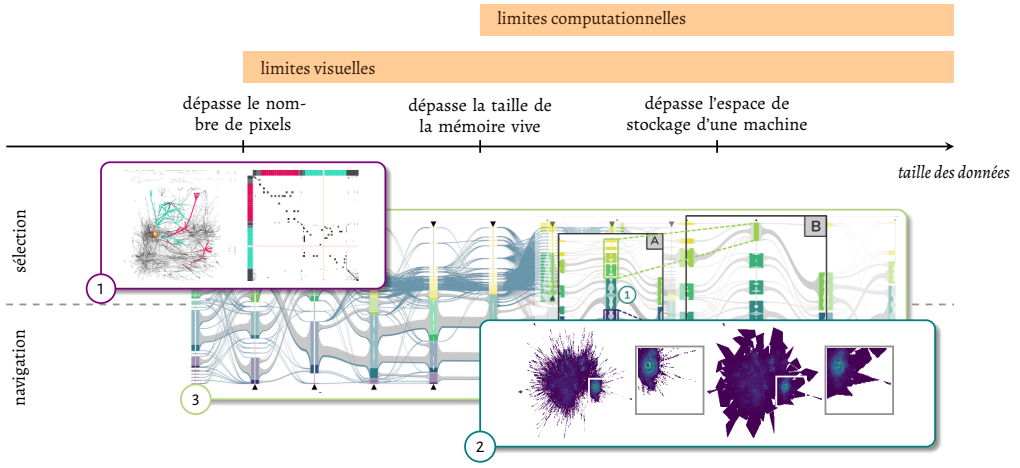


FIGURE 1.3 – Contributions de cette thèse (① CORFISH, ② COREDEM, ③ HIEPACO) relativement aux interactions principales étudiées (sélection et navigation) et aux problématiques de passage à l'échelle considérées (visuelles et computationnelles).

1.2 Axes de recherche et contributions

Dans cette thèse, nous nous attacherons à identifier des approches pour le passage à l'échelle de techniques de visualisation exploratoire pour les jeux de données grands ou complexes. Nos recherches sont motivées par les deux verrous scientifiques des limites visuelles et computationnelles et nous considérons trois axes de recherche qui sont : ① les vues multiples pour les données complexes, ② les cartes de densité pour les grands nuages de points et ③ les coordonnées parallèles agrégées pour les données multi-dimensionnelles. La figure 1.3 présente un aperçu de ces trois axes et la manière dont ils s'organisent par rapport aux deux verrous du passage à l'échelle.

Dans les vues multiples, l'espace écran est partagé entre plusieurs vues ce qui rend cet agencement très sensible à l'encombrement visuel. Dès lors, même en l'absence de limitations computationnelles concernant l'interactivité, l'efficacité d'interactions de surbrillance pour lier les vues entre elles est limitée visuellement. Nous proposons une méthode ① de déformation spatiale qui permet de ré-agencer les éléments graphiques pour compenser la limitation d'espace écran et agrandir les éléments d'intérêt pour permettre leur mise en évidence efficace. Nous montrons comment la méthode peut être appliquée à de multiples formes de représentations visuelles sur sept exemples.

Pour permettre l'analyse de données dont la taille dépasse les capacités en mémoire d'une machine standard ou même son espace de stockage, les systèmes de visualisation ont recours à des machines spécifiques qui sont nécessairement distantes du client de visualisation utilisé par l'utilisateur. Dans ce contexte, les temps d'interactions nécessitant un échange entre client et serveur sont fortement impactés par les temps de transfert, et la totalité des données ne peut pas être transférée en temps interactif, ni stockée dans l'espace mémoire de la machine client. Nous proposons une approche de compression géométrique ② pour réduire les données nécessaire à la reconstruction de la carte de densité de très grands ensembles de points à bas coût de stockage et transfert tout en permettant un degré d'interaction du côté client. Cette approche utilise un partitionnement des points pour constituer un ensemble de formes géométriques couvrantes pondérées qui est rendu de manière à préserver des propriétés élémentaires de la carte de densité. Nous implémentons plusieurs stratégies suivant cette approche et étudions leur efficacité à reconstruire une carte de densité fidèle à l'originale pour un budget de stockage donné. Nous montrons que plusieurs de ces stratégies sont comparables en efficacité à l'approche standard de *binning* tout en offrant l'avantage d'être non-régulières.

Pour permettre l'exploration de données multi-dimensionnelles massives sous la forme de coordonnées parallèle, nous utilisons l'abstraction par agrégation et l'utilisation d'une infrastructure de calcul distribué pour répondre aux deux verrous de scalabilité, visuelle et computationnelle , à la fois. L'abstraction par l'agrégation sert à la réduction de l'encombrement visuel, notamment pour l'interaction de sélection, en présentant des groupements d'éléments plutôt que des éléments individuels. Le degré d'abstraction de la représentation fournit également une garantie sur la taille des données échangées entre serveur et client à chaque interaction de sélection ou navigation. L'infrastructure distribuée permet la scalabilité computationnelle en pouvant faire face à la croissance des données par l'agrandissement de son réseau d'unités de calcul. Nous présentons un formalisme pour l'agrégation dans les coordonnées parallèles, une représentation abstraite focus+contexte ainsi qu'un ensemble d'interactions assurant la conservation d'un budget en éléments graphiques ③.

1.3 Organisation du document

Ce manuscrit de thèse est structuré en quatre chapitres principaux qui sont résumés dans les paragraphes suivant.

Le chapitre 2 (page 5) présente le contexte de la visualisation d'information, les limites visuelle et computationnelles liées à l'exploration de données de taille grandissante, et les différentes techniques et systèmes existants.

Le chapitre 3 (page 23) s'intéresse au cas particulier des vues multiples. Il présente dans un premier temps la problématique de scalabilité visuelle des interactions de mise en évidence dans ce contexte puis le principe de CorFish consistant à synchroniser entre plusieurs vues une déformation spatiale liée agrandissant les régions autour des entités d'intérêt pour l'utilisateur. Enfin, nous présentons l'application de cette technique à plusieurs types de représentations.

Le chapitre 4 (page 45) concerne la réduction de l'empreinte de stockage et de transfert de carte de densité pour l'exploration multi-échelle de grands ensembles de points dans le plan. Le chapitre présente les techniques existantes pour construire des cartes de densités à partir de données réduites et définit une forme de réduction géométrique *conservatrice* préservant la couverture et la masse de la carte de densité. Nous présentons ensuite un ensemble de stratégies composées de méthodes de partitionnement de données et de représentations agrégées communes qui sont implémentées et comparées, automatiquement et par une évaluation utilisateur, au regard de leur fidélité de reconstruction.

Le chapitre 5 (page 75) propose un système d'exploration de données multi-dimensionnelles massives sous la forme de coordonnées parallèles abstraites par agrégation permettant notamment des interactions de sélection de navigation multi-échelle. Le chapitre introduit un formalisme sous forme d'un graphe pour relier la structure d'une représentation abstraite des coordonnées parallèles aux données et décrire les interactions de sélection et navigation multi-échelle. Le système est présenté en deux temps : d'abord sous une forme non-hiérarchique exploitant le pré-calcul de plusieurs interactions, puis sous une forme hiérarchique reposant exclusivement sur du calcul à la volée. L'utilisabilité des systèmes est évaluée par des tests de performances et illustrée par une étude de cas.

Le manuscrit est ensuite conclu par un bilan et un résumé des perspectives de recherche envisagées. L'annexe A présente les détails des analyses statistiques conduites pour CorFish dans le chapitre 4. L'annexe B présente discute de deux interactions d'édition d'une hiérarchie pour HiePaCo non présentée dans le chapitre 5.

2 Visualisation interactive de grands jeux de données

Dans ce chapitre, nous proposons une introduction générale du domaine de la visualisation d'information interactive et plus particulièrement des techniques de visualisation qui sont les sujets de nos problématiques : les vues multiples, les cartes de densités et les coordonnées parallèles (section 2.1). Nous décrivons ensuite les ressources limitantes dans la visualisation interactive de grands jeux de données (section 2.2) et dressons un état de l'art des différentes approches de passage à l'échelle ayant été proposées (section 2.3).

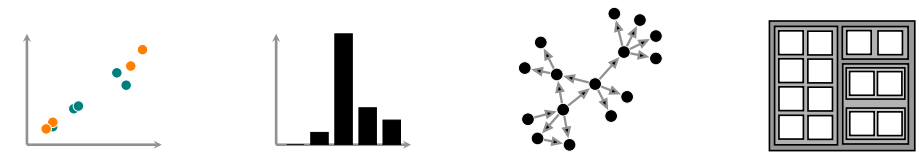
2.1 Visualisation d'information interactive

La visualisation regroupe les techniques et systèmes informatiques de représentation graphique et interactive de données utilisés pour en faciliter l'étude. C'est un domaine qui exploite conjointement les capacités computationnelles des ordinateurs et les capacités du système visuel humain qui est hautement parallèle et possède la plus grande bande passante parmi les centres cognitifs humains [165]. Les fondements de la visualisation d'information reposent sur l'exploitation efficace de cette liaison privilégiée pour transmettre précisément et rapidement l'information. On peut distinguer plusieurs niveaux de complexité entre les outils de visualisation qui dépendent de leur visée [35]. Les niveaux de complexité les plus bas sont ceux d'outils visant à *communiquer* une information à l'utilisateur ou à *observer* des données. Dans ce manuscrit nous nous intéressons au niveau de complexité supérieur correspondant à des techniques servant à analyser et explorer les données.

Une technique de visualisation *interactive* est composée d'une représentation graphique et de ses interactions. La représentation graphique repose sur un *idiome* [121] qui est un ensemble d'*encodages visuels* associés à un ensemble d'*interactions*. Au contraire de la visualisation scientifique qui s'intéresse à des données en général spatiales et décrivant des phénomènes continus, la visualisation d'information s'intéresse principalement à des données *discrètes*, composées d'un ensemble d'*entités* possédant des *attributs* ou des *relations*.

2.1.1 *Représentation graphique*

Une représentation visuelle encode les entités des données par des *marques*, entités graphiques unitaires, dont l'apparence est contrôlée par des *variables visuelles* telles que la position, la couleur,



(a) Nuage de points. (b) Diagramme à barres. (c) Diagramme nœuds-liens. (d) *Treemap*.

FIGURE 2.1 – Exemples de représentations visuelles. (a) Nuage de points pour neuf entités ayant deux attributs quantitatifs (encodé sur les abscisses et ordonnées) et un attribut catégoriel (encodé sur la teinte). (b) Diagramme à barres pour cinq entités possédant un attribut quantitatif. (c) Diagramme nœuds-liens pour dix entités où les relations entre entités sont représentées par un lien. (d) Treemap [90] pour un arbre à 14 feuilles où les relations *(parent,enfant)* sont représentées par l'inclusion.

etc, qui dépendent des attributs des entités des données [121]. Cette correspondance entre attributs graphiques et attributs des entités permet d'extraire des informations sur les données par le biais de reconnaissance de motifs et tendance sur la représentation graphique. La figure 2.1 présente quatre exemples de représentations visuelles : un nuage de points, un diagramme à barres, un diagramme nœuds-liens et une *treemap* [90]. Sur les deux premières, la position des entités est déterminée par des attributs alors que sur les deux autres elle est déterminée par les relations qui les relient.

Le choix d'une représentation visuelle dépend de différents critères. Le modèle *économique* de Wijk [168] demande à ce que les avantages apportés par la visualisation dans l'extraction d'information et de connaissances l'emportent sur les coûts computationnels et cognitifs induits par le processus de visualisation. Pour cela, deux critères sont à considérer : l'*expressivité* et l'*efficacité* des encodages visuels [109]. Une représentation est expressive lorsqu'elle retranscrit l'information recherchée et efficace lorsqu'elle utilise optimalement les capacités du support (écran) et du système visuel humain. Ainsi, le choix d'une variable visuelle est dirigé par l'importance de l'attribut encodé et sa nature, en général classifiée selon la nomenclature de Stevens [152] : *catégorielle* (ou *nominale*), *ordinale*, *intervalle* et *ratio*. Les types intervalle et ratio sont généralement réunis sous le terme *quantitatif*. Certaines variables visuelles suggèrent une échelle ou un ordre sous-jacent, comme la taille, la position ou la luminosité et sont donc adaptées aux attributs quantitatifs et ordinaux. Les variables identificatrices comme la forme ou la teinte, sont perçues comme toutes différentes et donc adaptées aux attributs catégoriels (cf. figures 2.2 et 2.3).

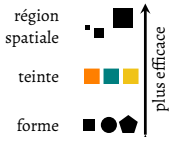


Figure 2.2: Efficacité de plusieurs variables visuelles pour les attributs catégoriels (d'après [121, p. 102]).

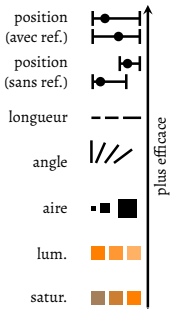


Figure 2.3: Efficacité de plusieurs variables visuelles pour les attributs quantitatifs et ordinaux (d'après [121, p. 102]).

L'efficacité d'une variable visuelle dépend aussi de la précision avec laquelle elle est perçue par les humains, de sa séparabilité à d'autres variables visuelles et de son effet de saillie. Le traitement *pré-attentif* [130] caractérise certaines propriétés visuelles comme la teinte, l'orientation ou la densité qui sont perçues très rapidement et sans parcours visuel de l'image, c'est-à-dire aussi rapidement quel que soit le nombre de « leurres » [165, chapitre 5]. De multiples travaux se sont attachés à la description de variables visuelles [15, 109] et à l'évaluation expérimentale de leur efficacité [109, 74]. Par exemple, pour les attributs ordinaux et quantitatifs, la position relativement à une échelle de référence a été déterminée la plus efficace, suivie par la position sans échelle de référence (cf. figure 2.3).

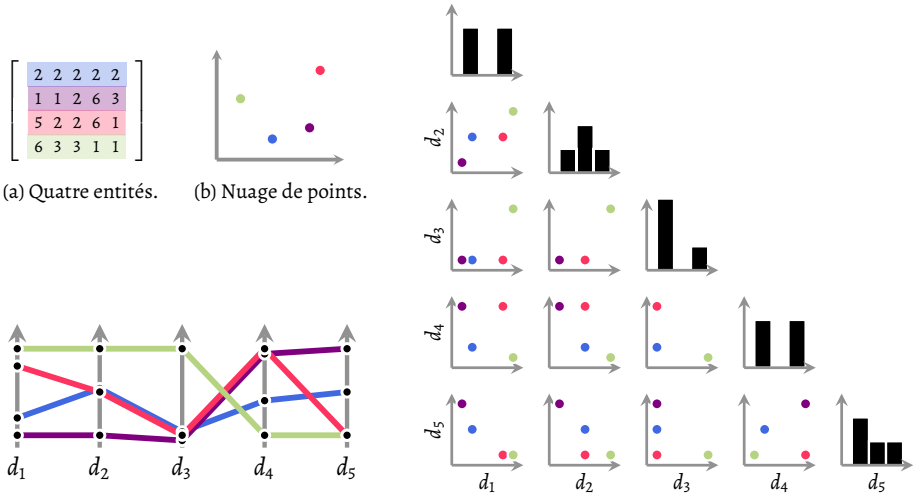
D'autres théories ont été présentées pour comprendre et évaluer l'efficacité d'une visualisation [154, 179, 95, 135]. Kindlmann et Scheidegger [95] présentent par exemple trois principes que sont la correspondance, la non-ambiguïté et la représentation de l'invariance pour évaluer la capacité d'une représentation visuelle à retranscrire des propriétés fondamentales des données. Enfin, a posteriori, des métriques ou mesures peuvent être utilisées pour évaluer une instance de visualisation [13] ou automatiser le choix de ses paramètres [115].

Nous prenons ici l'exemple des données multi-dimensionnelles, aussi appelées *multivariées* pour présenter différentes représentations visuelles pour les mêmes données, illustrées sur la figure 2.4. Des données multi-dimensionnelles s'apparentent dans le cas général à un tableau de valeurs soit une matrice, dont les lignes sont les entités des données et les colonnes leurs attributs (ou *dimensions*). Elles peuvent être représentées par exemple sous la forme d'un unique nuage de points, de coordonnées parallèles ou d'une matrice de nuages de points.

NUAGE DE POINTS Les multiples dimensions des données sont réduites à deux ou trois dimensions qui sont ensuite utilisées pour encoder par la position des entités sur un nuage de points. Dans l'exemple de la figure 2.4b, l'algorithme de réduction de dimension (t-SNE [108]) réduit les cinq dimensions à deux.

COORDONNÉES PARALLÈLES Les coordonnées parallèles sont un système de coordonnées multi-dimensionnelles introduit par Inselberg et Dimsdale [82] dans lequel une entité m-dimensionnelle est représentée par une ligne brisée coupant m axes parallèles, un pour chaque dimension, à la position correspondant à son attribut pour cette dimension. La figure 2.4c en présente un exemple pour quatre entités à cinq dimensions.

MATRICE DE NUAGE DE POINTS Matrice de nuages de points [38] correspondant aux couples de dimensions où toutes les paires de dimensions sont représentées. Sur la figure 2.4d



(a) Quatre entités. (b) Nuage de points.

(c) Coordonnées parallèles.

(d) Matrice de nuages de points.

FIGURE 2.4 – Techniques de visualisation pour des données multi-dimensionnelles. Pour illustrer leur encodage, les différentes techniques (b-d) présentent les mêmes données (a) et utilisent une teinte identique pour leurs marques correspondant à la même entité des données. Sur le nuage de points (b), les positions de chaque marque sont issues d'une réduction dimensionnelle des données. Dans les coordonnées parallèles (c), chaque entité est représentée par une ligne brisée dont les positions des sommets sur chacun des axes indiquent chacune la valeur d'un attribut de l'entité. Les matrices de nuage de points (d) représentent les entités des données par un nuage de points par couple de dimensions.

seule la partie inférieure de la matrice est dessinée pour éviter la redondance et la diagonale est utilisée pour présenter la distribution des entités sur chaque dimension sous la forme d'un histogramme.

Ces trois exemples illustrent différentes possibilités de représentations graphiques qui font chacune des compromis différents. Le nuage de points est par exemple la représentation la plus simple alors que les coordonnées parallèles et la matrice de nuage de points sont plus complexes à appréhender, car composées de multiples sous-vues. Pour cette raison également, ces vues sont en général accompagnées d'une interaction de mise en surbrillance permettant de faire le lien entre les sous-vues (cf. section 2.1.2). Coordonnées parallèles et matrice de nuage de points facilitent différentes tâches : la détection de points isolés ou aberrants est plus facilement accomplie sur les matrices de nuages de points alors que les coordonnées parallèles ont été montrées plus efficaces pour l'identification d'entité [32]. De plus, pour l'identification visuelle de corrélation c'est leur association qui est la plus efficace [32] (motifs illustrés sur la figure 2.5). Pour cette raison, ces idiomes peuvent être utilisés conjointement sur des vues juxtaposées constituant ainsi des vues multiples [145, 139], en général coordonnées par une interaction de type brossage et lien (cf. section 2.1.2).

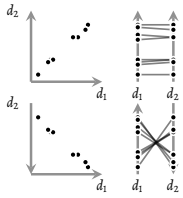
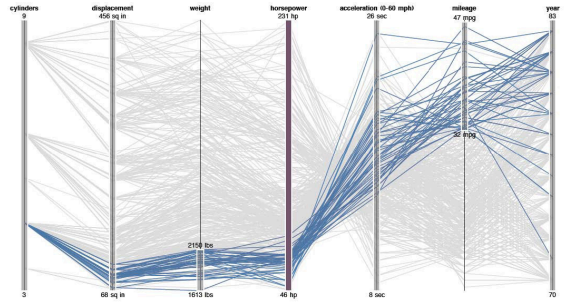


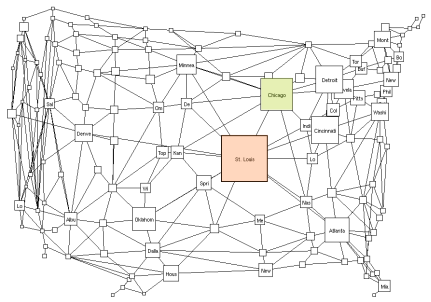
FIGURE 2.5: Motifs de corrélation (haut) et anti-corrélation (bas) en coordonnées cartésiennes (gauche) et parallèles (droite).

2.1.2 *Interactions*

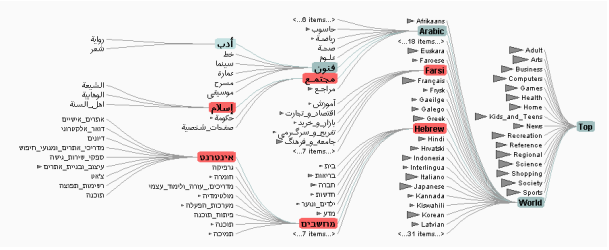
L'interaction est un aspect essentiel de la visualisation d'information [26, 76]. Sans, une technique de visualisation est réduite à une image statique ou animée automatiquement. Bien qu'une image statique possède de la valeur pour l'analyse et la transmission d'information [15], son utilité devient limitée pour des données complexes et à mesure que les données observées grandissent en nombre d'entités ou d'attributs. À partir d'une certaine taille ou un certain degré de complexité, lorsqu'une unique image ne peut plus contenir l'ensemble de l'information, l'interaction est un moyen pour l'utilisateur de pallier cette limitation en altérant l'image.



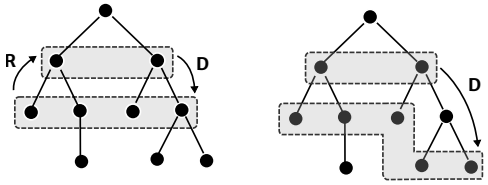
(a) Sélection composite (par intersection) sur les coordonnées parallèles issue du *brossage* d'une partie des 3e et 6e axes.



(b) Vue *fisheye* sur un diagramme nœuds-liens où le focus est le nœud . Les nœuds dans l'entourage topologique du focus sont aussi agrandis, avec un moindre facteur (p. ex.) [143].



(c) Vue focus+contexte d'un arbre avec de multiples foci. Les sous-arbres les moins pertinents sont cachés et symbolisés par ▷ [75].



(d) Illustrations des opérations de navigation *drill-down* (R) et *roll-up* (R) sur un arbre d'agrégation. Sur la gauche, le changement de niveau de détail est uniforme. Sur la droite, le changement de niveau de détail forme une zone de focus de plus haut niveau de détail (d'après [55]).

FIGURE 2.6: Exemples d'interactions de la catégorie « sélectionner » (a) et la catégorie « naviguer » (b-d).

On peut séparer les types d'interactions en deux catégories : les interactions de manipulation *indirecte* et celles de manipulation *directe*. Les techniques de manipulation indirecte correspondent aux interactions de l'utilisateur avec des menus, boutons ou curseurs externes à la vue. Les techniques de manipulation directe permettent à l'utilisateur d'altérer une représentation en manipulant directement les éléments de celle-ci. Les interactions de sélection sont l'essence même de ce principe : les entités sont encerclées ou brossées sur la vue et marquées immédiatement comme sélectionnées. Dans la définition BECKER et al. [10], une *interaction* est même définie plus restrictivement comme une manipulation directe associée à un changement immédiat.

Les systèmes de visualisation *exploratoire* suivent en général le mantra de SHNEIDERMAN [148] : « Overview first, zoom and filter, then details-on-demand ». Selon ce mantra, le processus exploratoire devrait commencer par la présentation d'une vue d'ensemble (*overview*) depuis laquelle l'utilisateur zoome sur des régions d'intérêt et en filtre les éléments. Si nécessaire, il est possible d'accéder au détail, à la demande (*details on demand*). Ensuite, l'exploration peut continuer vers des entités liées ou similaires ou encore redémarrer depuis la vue d'ensemble pour s'intéresser à une autre région ou question. Par ces successions de manipulations, l'utilisateur se forme un *modèle mental* des données en navigant d'un *focus* à un autre, c.-à-d. d'un sous-ensemble des données, d'une facette ou d'une question à une autre.

La taxonomie d'interactions de YI et al. [172] propose sept catégories orientées par les intentions de l'utilisateur qui sont : sélectionner, explorer, reconfigurer, encoder, abstraire/détailler, filtrer et connecter. Celle de HEER et SHNEIDERMAN [76] en propose douze, regroupées en trois grandes catégories qui sont (1) la spécification des données et de la vue, (2) les manipulations de la vue et (3) le procédé et la provenance. Nous détaillons dans ce qui suit trois types d'interactions de la catégorie manipulation de la vue qui sont pertinentes pour ce manuscrit.

Sélectionner

Les interactions de sélection, aussi appelées *brossage* [9] dans certains cas, servent à l'utilisateur à marquer des entités par exemple pour en conserver la trace au cours d'un ré-arrangement. Une

sélection définie, ou en cours de définition, est en général indiquée par un *retour visuel immédiat* sous la forme de mise en surbrillance de la région ou des entités graphiques ciblées.

Les modes de sélection les plus communs sont le survol du pointeur, le clic, la sélection par région (p. ex. rectangulaire ou « lasso ») ou d'autres curseurs spécifiques parfois appelées *pinceau* (*brush*), comme le pinceau de sélection angulaire illustré sur la figure 2.7. Les modes de sélections sont parfois étendus par des opérateurs d'altération ou composition de sélections, dont les plus simples sont les opérations ensemblistes que sont l'union, la soustraction et l'intersection. La sélection fonctionne souvent comme une étape préliminaire à une seconde action, définissant le(s) élément(s) d'entrée de cette dernière. L'action suivante peut être la mise en surbrillance par une teinte distinctive (*highlighting*), le filtre, la demande de détail, etc. La figure 2.6a présente un exemple de mise en surbrillance d'entités sur des coordonnées parallèles résultant de la sélection par brossage des entités de basse valeur sur le 3e axe et de hautes valeurs sur le 6e axe, avec ces deux actions consécutives composées par intersection.

Les interactions de sélection sont reliées au principe de la mise en surbrillance qui est notamment relié à la mise en évidence et à l'interaction de brossage et lien.

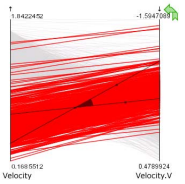


FIGURE 2.7: Pinceau de sélection angulaire dans les coordonnées parallèles [71].

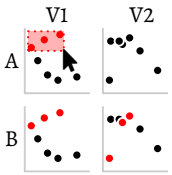


FIGURE 2.8: Brossage par sélection rectangulaire (A) et lien par surbrillance (teinte) (B) sur deux nuages de points (V1 et V2).

MISE EN ÉVIDENCE La surbrillance et le filtre sont des cas particuliers d'interactions de *mise en évidence* dont l'objectif est plus généralement de rendre visuellement *saillante* une région ou une sélection d'entités. HALL et al. [69] fournissent une formalisation et un état de l'art des approches utilisées pour provoquer cet effet. Ils distinguent trois catégories : la suppression ou filtre, les approches par grossissement (altération de la taille et de la position) et les approches qui altèrent d'autres variables visuelles que la taille et la position comme la teinte.

BROSSAGE ET LIEN Le brossage est communément utilisé avec la surbrillance dans les interactions de brossage et lien (*brushing & linking*), aussi appelées *liens interactifs*, qui consiste à coordonner plusieurs vues en mettant en surbrillance les entités brossées par l'utilisateur sur l'une sur toutes les autres. Cette interaction est illustrée sur deux nuages de points sur la figure 2.8 et détaillée en contexte en page 10.

Naviguer

La visualisation fonctionne souvent par le principe de *fenêtre* sur un espace, fenêtre dont la taille et la position sont modifiées par des interactions de déplacement de caméra (ou *pan*) et de zoom, dit géométrique. Le *zoom sémantique* [11] modifie à la fois la quantité de données présentées et leur mode de représentation en fonction du niveau de zoom, révélant plus de détails sur la région *focus*. Cette même idée a donné naissance aux méthodes *focus+contexte* [39] qui présentent, sur une même vue, les régions de focus à un haut niveau de détail et les régions de contexte à faible niveau de détail [63].

VUES *FISHEYE* Lorsque la différence de niveau de détail perçue est due à une différente de taille et de positionnement des entités graphiques, ces méthodes sont aussi appelées vues *fisheye* ou représentations par *déformation* [19]. Ces exemples de vues focus+contexte opèrent généralement par déformation de l'espace de représentation et plus spécifiquement en grossissant les régions ou entités focus au détriment des autres (cf. figure 2.6b).

FOCUS+CONTEXTE DANS L'ESPACE DES DONNÉES Les vues focus+contexte dans l'espace des données consistent en une déformation qui concernent soit le niveau d'agrégation (contexte plus agrégé [127, 55]), soit la visibilité (contexte caché [75]). Pour ces approches, des fonctions de *degré d'intérêt* (DOI [1]) [63] sont définies sur les données et calculent l'importance de chaque entité relativement à sa proximité aux entités ayant été sélectionnées par l'utilisateur. Sur la figure 2.6c par exemple, la fonction de DOI permet de déterminer quelle portion de la hiérarchie est « repliée » au profit des nœuds proches des nœuds foci. Sur la figure 2.6b, c'est la fonction de DOI dans l'espace topologique qui détermine le degré de grossissement des nœuds. Par extension, le terme focus+contexte est aussi

1. degré d'intérêt (*degree of interest*)

FIGURE 2.9 – Deux sélections à l'œuvre définies par l'outil MYBRUSH [97] sur un agencement de vues multiples représentant des données d'un jeu vidéo de football. La sélection de couleur verte correspond aux joueurs de milieu de terrains (définie via le diagramme à barres). La sélection aux contours jaunes correspond aux joueurs avec de fortes capacités d'attaques (définie via les coordonnées parallèles) .

utilisé pour d'autres interactions utilisant une fonction de DOI pour appliquer de la mise en surbrillance non-binaire [46].

ABSTRAIRE/DÉTAILLER Les représentations agrégées suivant une hiérarchie sont associées à un couple d'interactions de navigation de demande de détails et d'abstraction appelées respectivement *drill-down* (détailler) et *roll-up* (abstraire) [55, 62]. Ces deux interactions gouvernent le niveau de détail auquel est représenté un ensemble d'entités ou toutes les entités. Le *drill-down* déplace le niveau de détail plus profondément dans la hiérarchie d'agrégation alors que le *roll-up* fait l'opération inverse. La figure 2.6d présente deux illustrations de ces opérations, avec un exemple de niveau d'agrégation inégal formant une vue focus+contexte.

Coordonner

Comme mentionné précédemment, lorsque les données sont complexes l'utilisation de plusieurs idiomes de visualisation sous la formes de *vues multiples coordonnées* [138, 145, 139] permet une analyse selon différentes perspectives. Les interactions de brossage et lien ou lien interactif (*brushing & linking* [9]) sont l'exemple canonique de coordination entre des vues multiples. Elles consistent à permettre la sélection d'entités par *brossage* sur une vue et la mise en surbrillance des données correspondantes sur toutes les autres vues par un encodage visuel commun.

La figure 2.9 présente un agencement de quatre vues pour l'analyse de données d'un jeu vidéo de football avec deux sélections actives. La première teinte les entités graphiques correspondant aux joueurs de milieu de terrain en vert. Elle a été définie sur la vue par diagramme à barres qui représente le nombre de joueurs pour chaque position. La seconde entoure de jaune les entités graphiques correspondant aux joueurs aux meilleures capacités d'attaque. Elle a été définie par brossage du premier axe sur la vue par coordonnées parallèles qui représente sept attributs des joueurs. Dans ces exemples l'effet de *lien* fonctionne par encodage partagé entre plusieurs vues de différentes formes. Les différences entre vues peuvent présenter une difficulté pour la concep-

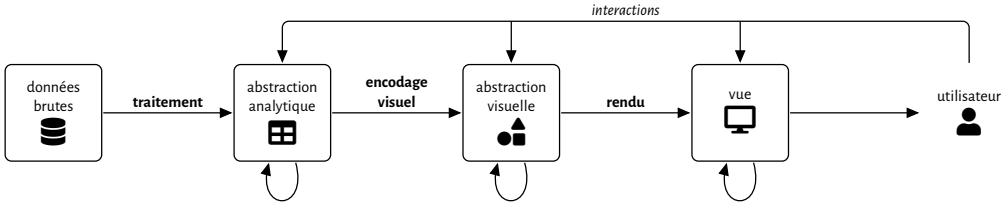


FIGURE 2.10 – Schéma du *pipeline* de visualisation de référence (d'après CARD et al. [26]). Ces composants peuvent servir à décrire un grand nombre de techniques de visualisation et d'applications.

tion d'interactions et BALDONADO et al. [6] préconisent de conserver, au cours des interactions, des états cohérents entre vues partageant des données (p. ex. zoom, surbrillance) ainsi que des *potentialités d'interaction* (*affordances*), c.-à-d. actions d'interactions suggérées, cohérentes entre vues.

2.1.3 Modèles d'application de visualisation

Les modèles de flot de données (ou *pipeline*) de visualisation sont des schémas permettant à la fois de décrire et d'implémenter des outils de visualisation interactive [37, 141] mais aussi d'expliquer les différents niveaux d'implication de l'utilisateur dans le processus de construction d'une représentation [26]. Des données à la représentation graphique, le processus de conception d'une visualisation a de nombreux degrés de liberté et variations qui sont déterminants pour la représentation finale alors même que l'espace de ses possibilités est immense [121]. L'optimisation automatique de ce processus est difficile et c'est une des raisons pour laquelle l'interaction utilisateur est cruciale.

Modèle de référence

Le pipeline de visualisation présenté en figure 2.10 décrit le processus de visualisation en quatre étapes. La *transformation des données* inclut la structuration des données brutes, le filtrage des entités ou attributs d'intérêt, ou encore l'agrégation. À cette étape, un exemple d'interaction utilisateur est le changement d'un paramètre d'agrégation ou d'un filtre. L'application en amont de ces opérations permet de réduire la taille des données le plus tôt possible et d'éviter des calculs inutiles. L'étape d'*encodage visuel* est au cœur du pipeline et consiste en la transformation des données tabulaires en *abstraction visuelle*, c.-à-d. primitives graphiques. À cette étape, des exemples d'interaction utilisateur sont la sélection et le réagencement des entités graphiques (p. ex. tri). Enfin, l'étape de *rendu* consiste à la construction de la vue, c.-à-d. de l'image présentée à l'utilisateur, et inclut donc notamment un découpage de l'espace graphique visible (*clipping*) et une rastérisation. À cette étape, les interactions utilisateur principales sont le zoom et le déplacement qui altèrent la position et taille de la fenêtre de visualisation.

Du point de vue de l'implémentation, permettre des interactions peut complexifier la conception et structuration d'une application informatique. La boucle d'interaction du pipeline de visualisation correspond en général à une boucle de traitement d'entrée (p. ex. clavier ou souris) dans l'implémentation. Plusieurs travaux ont proposé des modèles facilitant la conception ou l'implémentation efficace de tels systèmes (p. ex. automate à états [3], architecture multi-processeurs [134]).

Modèle client-serveur

Pour des raisons de centralisation de données, d'environnement mobile ou de besoin de performances spécifiques, de nombreux systèmes adoptent une architecture client-serveur où les composants du pipeline de visualisation sont séparés en deux parties prises en charge par deux

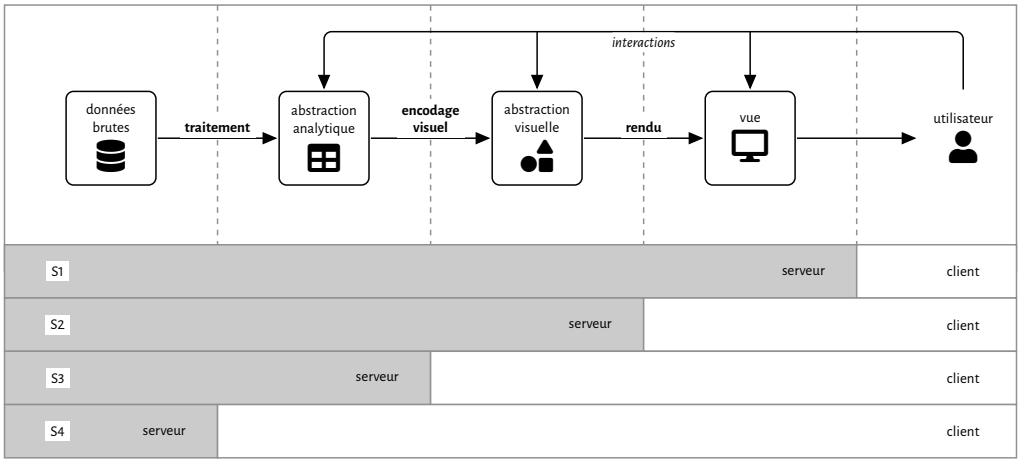


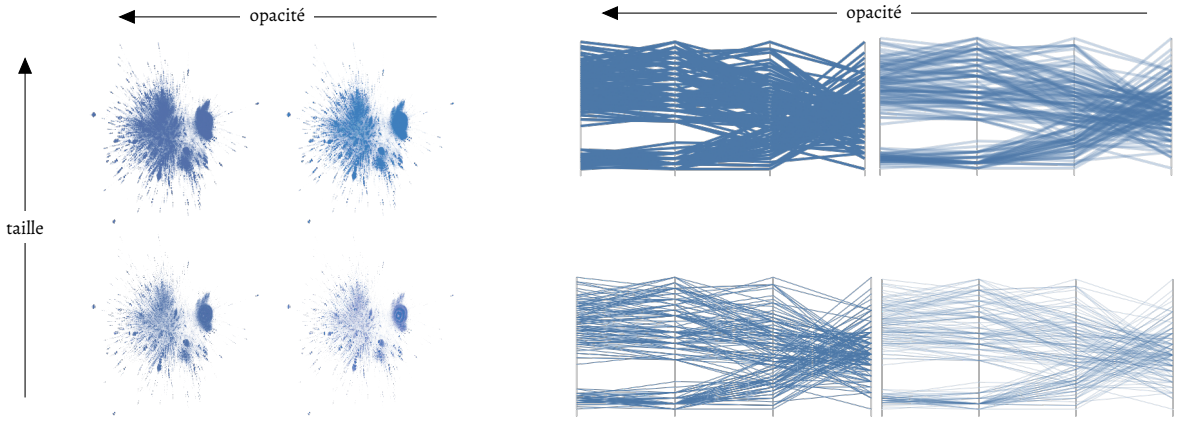
FIGURE 2.11 – Quatre stratégies de pipeline de visualisation déportée. Suivant la stratégie, le coût et la fréquence des transferts réseau et les besoins de ressources sur le client varient. Plus la séparation est en aval du pipeline, plus les échanges entre client et serveur seront important durant les interactions.

entités reliées par une liaison réseau. Client et serveur ont en général différentes ressources de sorte que la capacité mémoire et la puissance de calcul du serveur sont supérieures à celles du client. Dans ce type d'architecture, les interactions sur le client de visualisation sont traduites en *requêtes* au serveur. Les latences d'interaction du système dépendent d'une part des capacités de calcul du client et du serveur (relativement aux données qu'ils traitent), et d'autre part de la latence et de la bande passante de la liaison réseau (relativement aux données transférées). Le pipeline peut être divisé entre client et serveur à différents stades comme illustré sur la figure 2.11, ce qui détermine la nature des données transitant sur le réseau ainsi que les responsabilités du client.

Selon la stratégie S4, le client reçoit les données brutes et éventuellement une spécification de la visualisation dans un échange initial, et toutes les interactions peuvent s'effectuer ensuite sans transfert réseau supplémentaire. Cette stratégie n'est cependant pas utilisable lorsque les ressources en mémoire et calcul du client sont insuffisantes pour traiter la taille des données brutes.

À l'opposé, selon la stratégie S1, le client reçoit des données s'apparentant à des images. L'avantage principal de cette approche est que la quantité de données transitant, comme celle manipulée localement par le client, est indépendante de la complexité ou taille du jeu de données original. Les coûts de transfert et de traitement du côté client sont donc fixes, même lorsque la taille des données analysées augmente. Cependant, cette configuration laisse très peu, voire aucune, interaction possible sans échange avec le serveur ce qui est problématique pour la réactivité du système.

Enfin, les stratégies S2 et S3 sont des compromis où les données transférées sont intermédiaires dans le pipeline et donc en général de taille réduite. Dans LiVID [133] par exemple, le client reçoit un échantillon pondéré des données à partir duquel est rendue une carte de chaleur représentant une estimation de la fonction de densité de ces données. MORITZ et al. [118] présentent une architecture déterminant dynamiquement, à chaque interaction, la stratégie minimisant les latences du réseau. Lorsque la vue présente des données de grande taille, le serveur traite les étapes de filtrage et d'agrégation alors que pour une vue de données réduites, ces étapes sont traitées par le client supprimant ainsi les transferts réseaux pour les interactions modifiant le filtrage. Réduire ainsi les transferts via le réseau sert à réduire les latences d'interaction, mais ménage également la bande passante du client et peut permettre dans certains cas de maintenir une forme d'interactivité lors de pertes temporaires de connexion.



(a) Nuages de points.

(b) Coordonnées parallèles.

FIGURE 2.12: Effet de l'altération (en taille et opacité) de l'apparence des marques sur (a) un nuage de points (jeu de données `panama_papers`, ≈*743 000* points) et sur (b) des coordonnées parallèles (jeu de données `iris`, *150* entités).

2.2 Défis et passage à l'échelle

La conception d'un système de visualisation efficace nécessite la considération de trois principales limitations : les ressources computationnelles, le système de perception et cognition humain et les écrans, supports de la visualisation. Ces trois types de limites sont souvent présentées sous la forme de deux difficultés fondamentales : une difficulté *perceptuelle* (ou *visuelle*) et une difficulté d'*interaction* (ou *technique*, *computationnelle*) [55, 106, 60, 66]. La première concerne l'utilisabilité de la représentation graphique, c'est-à-dire sa capacité à transmettre efficacement l'information contenue dans les données (cf. section 2.1). Cette difficulté concerne matériellement les écrans, dont la définition (en nombre de pixels) limite la quantité d'information affichable simultanément, et les capacités cognitives et perceptuelles du côté utilisateur. La seconde concerne les limites en ressources de mémoire et de calcul, communes à toute application informatique. Les limites de temps de calcul sont particulièrement problématiques pour le maintien de la réactivité du système et notamment des interactions dont dépendent, en partie, les performances cognitives de l'utilisateur [106].

2.2.1 *Limites visuelles*

Un écran standard possède aujourd'hui une définition comprise entre un million (HD) et huit millions (4K UHD) de pixels (cf. tableau 2.1). La limitation de l'espace écran en nombre de pixels, problème parfois appelé le *problème de l'immobilier de l'écran* (*screen real-estate problem*), limite fondamentalement le nombre de marques simultanément visibles. Ainsi, lorsque le nombre d'entités est assez grand, il est impossible de concevoir une vue d'ensemble des données utilisant une marque par entités, même en réduisant la taille des marques à un seul pixel [43]. Maximiser l'utilisation de cette ressource permet de minimiser le recours aux interactions de navigation, mais peut cependant produire un excès d'information néfaste à l'interprétation efficace de la vue.

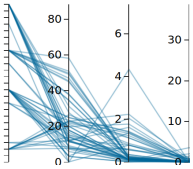
	DÉFINITION
	≈ 1 – 3
	≈ 3 – 4
	≈ 2 – 8

TABLE 2.1: Définition standard des écrans de différents périphériques en millions de pixels.

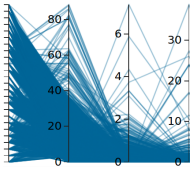
L'encombrement visuel (*visual clutter* [52]) désigne le phénomène de perte d'information induit par le recouvrement, le chevauchement et la concentration d'entités graphiques. Ce phénomène concerne les représentations denses et survient rapidement lorsque la taille des données croît. Il existe plusieurs approches pour remédier à l'encombrement visuel [52]. La diminution de la taille des marques ou l'utilisation de la semi-transparence pour dévoiler les marques recouvertes sont deux approches communes, illustrées sur la figure 2.12. Cependant, la diminution de la taille est limitée par la définition et peut perturber la perception d'autres variables visuelles (forme, couleur) jusqu'à les rendre imperceptibles. La composition de marques semi-transparentes est en

TYPE	ÉCHELLE (S)	EXEMPLES
traitement de perception	*0,10*	reconnaître un motif, suivre une animation
réponse immédiate	*1*	clic sur un lien, sélection d'un objet
tâche unitaire, brève	*10*	modifier une ligne de texte, déplacer une pièce aux échecs

TABLE 2.2 – Échelles temporelles des temps de réponse humain aux interactions (d'après CARD et al. [27]).



(a) *109* entités.



(b) *7637* entités.

FIGURE 2.13: Coordonnées parallèles pour différentes nombres d'entités (opacité : 70 %).

général limitée à *256* différents niveaux de transparence par l'implémentation, mais en pratique uniquement utile jusqu'à cinq marques superposées [72].

Certains idiomes sont plus exposés à l'encombrement visuel. Les coordonnées parallèles par exemple ne représentent efficacement que quelques centaines d'entités. Lorsque trop d'entités sont représentées, les lignes se croisent et se chevauchent résultant en de la perte d'information. Sur l'exemple de la figure 2.13, la vue à *7637* entités ne transmet que les valeurs extrêmes de chaque dimension et cache les éventuels motifs, corrélations et la distribution des entités sur chaque dimension. Les vues multiples sont un autre exemple d'idiome sujet à l'encombrement visuel. Dans ce cas-ci, l'espace écran et donc le nombre de pixels est partagé entre les vues. Par conséquent, le problème d'encombrement visuel survient, sur chaque vue, pour des quantités d'entités plus faibles que sur une vue simple. Pour des écrans d'environ un million de pixels et une douzaine de vues, le nombre maximal d'entités représentables efficacement par vue se situe sous mille entités [121]. Lorsque le nombre de marques est supérieur au nombre de pixels, il doit être réduit par agrégation et le détail des entités n'est rendu disponible que par des interactions de navigation (p. ex. zoom, navigation hiérarchique, cf. section 2.1.2). L'agrégation utilisée dans une vue d'ensemble peut modifier son encodage visuel par rapport à celui des vues détaillées résultant en une approche semblable à celle du zoom sémantique.

Les capacités de mémoire et d'attention humaines sont aussi des limitations visuelles à prendre en compte dans ces changements. Cela concerne notamment les objets non-visibles ou les objets qui ne sont pas le sujet de l'attention qui peuvent être l'objet de cécité au changement (*change blindness*) [149]. Les limitations de la mémoire à court terme ou *mémoire de travail* encouragent par exemple à privilégier la juxtaposition à l'animation pour la comparaison [121] et la mise en évidence explicite d'éléments d'intérêt par l'utilisation de variables pré-attentives sert à faciliter la focalisation de l'attention [73].

2.2.2 *Limites computationnelles à l'interactivité*

La principale difficulté de la conception de systèmes de visualisation pour de grands jeux de données est la réactivité, c.-à-d. le maintien des latences de réponses aux interactions de l'utilisateur à des taux non bloquant pour l'utilisateur. La réactivité est compromise par les limites matérielles des ordinateurs et autres infrastructures informatiques (grappe de machines, liaison réseau) et la taille grandissante des données visualisées. La définition d'un jeu de données *grand* et *limitant* pour les performances d'application de visualisation a évolué avec les composants informatiques et plus généralement des capacités en mémoire et en puissance des unités de calcul. En 1987, les nuages de points de *50 000* points étaient problématiques pour les temps de rendu [31], en 2002 les cartes graphiques modernes permettent de rendre un million de points interactivement alors qu'en 2013 le système IMMENS [107] permet des interactions de brossage et lien sur un milliard de points en conservant une réactivité de 50 images par seconde. Les sources de latences dans les systèmes de visualisation interactifs peuvent être le temps de traitement ou de requêtes à des systèmes de base de données, les temps de transfert réseau et ou encore les temps de rendu des vues.

Nécessité de réactivité

L'impératif de réactivité provient de la volonté de permettre l'analyse des données « à des rythmes en résonance avec le rythme de la pensée humaine » [76], de sorte que l'outil d'analyse soit en quelque sorte une extension du système cognitif. Ce rythme de pensée humaine varie suivant les tâches accomplies : le modèle présenté par CARD et al. [27] distingue trois catégories de tâches cognitives opérant à différentes échelles temporelles (cf. tableau 2.2). Selon CARD et al. [27], les tâches de brossage appartiennent par exemple à la catégorie de traitement de perception ce qui suggèrent qu'elles nécessitent un retour visuel sous *100 ms* environ. D'autre part, LIU et HEER [106] ont montré expérimentalement que des latences additionnelles de *500 ms* affectent négativement et durablement les performances cognitives des utilisateurs pour les interactions de déplacement et de brossage et lien comparativement à des conditions à faible latence. Ces résultats ne s'appliquent en revanche pas à l'opération de zoom.

Supports de calcul

Les latences de temps de traitement apparaissent lorsque les traitements (partitionnement, algorithme de dessin, etc) sont complexes ou que les données sont très grandes. Les processeurs modernes (CPU [2] et GPU [3]) sont composés de multiples unités de calcul qui permettent d'effectuer des calculs simultanément (calcul *parallèle*) pour améliorer les performances des traitements. On distingue deux grandes formes de parallélisme. Le parallélisme des *tâches* consiste à exécuter de manière concurrente différentes tâches, c.-à-d. des parties indépendantes d'un algorithme. Le parallélisme des *données* divise les données en tranches indépendantes et chaque unité de calcul exécute les mêmes tâches sur une tranche. À la fin de l'exécution, une étape de collecte réunit les sous-résultats. Cette approche est particulièrement adaptée aux problèmes homogènes traitant de grandes données, mais peut devenir inefficace pour des cas irréguliers. Le parallélisme du pipeline [117, 160] est une troisième forme consistant à assigner un module du pipeline à chaque unité de calcul et à traiter les données en flux.

L'utilisation efficace du parallélisme nécessite d'adapter les algorithmes et architectures traditionnelles de visualisation. Cette adaptation représente des difficultés car la parallélisation requiert certaines propriétés comme l'existence de tâches indépendantes (parallélisme de tâche) ou la capacité de l'algorithme à pouvoir traiter les données morceau par morceau dans un ordre quelconque (parallélisme des données). L'efficacité du parallélisme dépend du niveau de *concurrence* (quantité de traitements parallèles). Le parallélisme de tâches est limité en pratique au nombre de tâches indépendantes existantes. Le parallélisme de pipeline tend à résulter en plus de *concurrence* que le parallélisme de tâches mais reste limité au nombre de modules (en pratique pas plus de dix [117]). Le parallélisme des données est plus communément utilisé en visualisation car il tend à être bien équilibré dans ce contexte et exhibe un bon niveau de concurrence pour de grands jeux de données [117].

Les machines standards ont typiquement jusqu'à *64 Go* de mémoire CPU et jusqu'à *4 Go* de mémoire GPU, avec la bande passante de la mémoire de la première de l'ordre de quelques dizaines de *Go/s* et de la seconde de la centaine de *Go/s*. L'augmentation des ressources matérielles d'une machine (ajout de mémoire, remplacement de processeur) est une manière d'étendre ces limites jusqu'à un certain point. La capacité mémoire pourra être multipliée par *100* mais les mises à jour seront rapidement très coûteuses. Deux autres solutions sont possibles : utiliser des algorithmes à mémoire externe (*out-of-core*) [159, 42] et une infrastructure de traitements distribués.

Les algorithmes à mémoire externe sont conçus pour ne traiter qu'une fraction des données à la fois d'une manière similaire au traitement en flux (*streaming*). Cette approche rend possible le traitement de données plus grandes que la mémoire disponible mais peut souffrir des lenteurs de la lecture depuis le disque (débit de lecture inférieur à *3 Go/s* pour un SSD [4]).

2. processeur (*central processing unit*)
3. processeur graphique (*graphics processing unit*)
4. d'après `https://ssd.userbenchmark.com/`

Les infrastructures distribuées de stockage et calcul sont des grappes de machines (*cluster*) inter-connectées par une liaison réseau sur lesquelles les données sont réparties et les calculs ou requêtes s'effectuent en parallèle (parallélisme des données). Ces infrastructures bénéficient de la possibilité d'augmenter les ressources matérielles à bas coût en ajoutant des machines supplémentaires à la grappe. La mémoire est distribuée entre les unités de calcul ce qui rend la communication entre unités de calcul très coûteuse. Ces supports de calcul sont donc plus efficaces pour des problèmes peu *couplés*, par exemple ceux dont les données sont parallélisables. *MapReduce* [45] est un paradigme versatile de traitement par lots utilisé sur les infrastructures distribuées et Spark [173, 174] un moteur de calcul alternatif faisant une utilisation intensive de la mémoire vive pour s'affranchir autant que possible des surcoûts de lecture depuis les disques lors de traitements itératifs.

Temps de transfert

	TYPE	DÉBIT	LAT.
mobile	2G	*0,1*	500
	3G	*8*	100
	4G	*90*	50
filaire	ADSL	*7*	60
	THB	*90*	30

TABLE 2.3: Débit descendant (*Mbit/s*) et latence (*ms*) *typiques* pour différents types de liaison.

Les grandes échelles de jeu de données requièrent pour leurs traitements des serveurs ou infrastructures spécifiques, qui sont en général distants du client de visualisation. Le pipeline de visualisation est alors divisé (cf. section 2.1.3) et certaines interactions nécessitent un échange de données entre client et serveur (requête/réponse). Cet échange induit un délai qui peut dépendre de deux facteurs, en général hors du contrôle des concepteurs, la latence et la bande passante de la liaison réseau.

Une connexion 4G (LTE Cat. 4) a par exemple un débit maximum descendant (respectivement ascendant) de *150 Mbit/s* (respectivement *50 Mbit/s*). Dans les faits cependant, les débits effectifs sont plus bas (cf. tableau 2.3). Le débit moyen mobile dans le monde au premier trimestre 2017 allait de *26,0 Mbit/s* au Royaume-Uni à *2,80 Mbit/s* au Venezuela, avec *17,4 Mbit/s* en France. À la même période, le débit moyen pour les connexions filaires était de *10,8 Mbit/s* en France et *7,2 Mbit/s* à l'international [2]. Réduire et contrôler la taille des données échangées entre client et serveur est donc essentiel pour réduire les latences pendant l'interaction, et peut également permettre de sauvegarder la bande passante de la connexion ascendante du serveur ainsi que celle de la connexion descendante du client.

Temps de rendu

Les processeurs graphiques (GPU) peuvent être utilisés pour paralléliser efficacement l'étape de rendu de la visualisation et ainsi assurer un rendu rapide. La rapidité du rendu est notamment importante pour la fluidité des animations et les retours visuels immédiats. Elle est mesurée en nombre d'images produites par le système par seconde (en *i/s* pour *images par seconde*) avec une valeur raisonnable à *10 i/s* [20, 58] qui est la vitesse de défilement nécessaire pour que les humains perçoive une *animation* plutôt qu'une succession d'images indépendantes [21].

L'interface de programmation OpenGL, et son homologue dans le navigateur WebGL, servent à utiliser l'accélération graphique pour maintenir ce taux de rendu dans la visualisation de plus d'un million d'éléments graphiques. L'interface de programmation sert aussi à faciliter l'utilisation de certaines techniques de rendu comme le rendu de marques semi-transparentes (*alpha blending*) ou l'anti-crénelage (*anti-aliasing*). L'efficacité des rendus parallèles au GPU atteint ses limites pour de grands nombres d'éléments dans certains cas où la parallélisation est déséquilibrée ou lorsque la complexité de rendu est importante comme pour le calcul coûteux tel que celui de gaussiennes [133].

2.2.3 Scalabilité

La tendance continue d'augmentation de la taille des jeux de données est conduite par de multiples facteurs. Les améliorations dans l'acquisition de données et les technologies de capteurs conduisent à une génération de plus en plus importante de données dans des environnements informatisés telle que la simulation et la journalisation mais aussi l'expansion de la place des

infrastructures et outils informatiques dans la vie quotidienne. Les systèmes de visualisation sont donc voués à être utilisés pour explorer et analyser des jeux de données de taille supérieure à celle pour laquelle ils ont été conçus.

Cette tendance est incitative à la conception de systèmes possédant des mécanismes d'adaptation, permettant de prévenir et compenser les problématiques présentées ci-dessus. On utilisera l'anglicisme « *scalabilité* » (en français *passage à l'échelle* ou *extensibilité*) pour désigner cette capacité à s'accommoder d'une augmentation de charge en conservant une certaine efficacité. Dans notre sujet d'étude, la charge correspond à la quantité de données en nombre d'entités. Par extension, une technique sera *scalable* si elle a cette aptitude. Le terme *scalable* est par ailleurs aussi utilisé dans la communauté de visualisation d'information pour désigner la capacité à être efficace face à des échelles de données considérés grandes pour l'époque, sans pour autant mentionner de mécanisme adaptatif.

2.3 Techniques et systèmes de visualisation *scalables*

Dans cette section, nous présentons différents moyens utilisés pour résoudre les problèmes de passage à l'échelle décrit dans la section précédente. Ces techniques sont présentées selon les deux aspects de la scalabilité visuelle et computationnelle.

2.3.1 Scalabilité visuelle

Ellis et Dix [52] ont établi une taxonomie de techniques permettant de réduire l'encombrement visuel pour la visualisation dont nous présentons une partie sur le tableau 2.4, accompagnée de références aux travaux mentionnant ou employant ces techniques pour les nuages de points et les coordonnées parallèles. Sept techniques sont distinguées et groupées en trois catégories : les techniques altérant l'apparence des marques (taille ou opacité), celles déformant leurs positions et enfin les techniques d'*abstraction* par l'échantillonnage ou l'agrégation (spatiale ou non). Les deux critères de comparaison majeurs de ces techniques sont la capacité à éviter la superposition (1) et la scalabilité (2), c.-à-d. la capacité à réduire l'encombrement visuel même lorsque le nombre d'entités des données augmente. Trois autres critères sont reportés pour mettre en évidence les compromis de chaque technique : la conservation de l'information spatiale des entités (3), la possibilité de présenter un ou plusieurs autres attributs des entités (p. ex. par la teinte) (4) et la capacité à montrer la densité des entités se superposant (5).

Altération de l'apparence des entités graphiques

Comme mentionné dans la section précédente, une méthode simple pour réduire l'encombrement visuel est d'altérer l'apparence des entités en changeant leur taille ou opacité. Réduire la taille (A1) peut diminuer la superposition entre les marques alors que la semi-transparence (B1) permet de révéler les marques recouvertes. Ces deux solutions deviennent inefficaces à partir d'un certain nombre d'entités à cause de limites de la définition des écrans ou de l'acuité visuelle (A2, B2). La réduction de la taille et de l'opacité interfèrent également avec la perception d'autres variables visuelles des marques (p. ex. teinte, forme) (A4, B4).

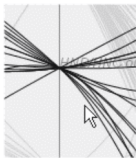
Déformation spatiale

Parmi les techniques de déformation spatiale, on distingue celles (C) ajoutant un déplacement aux entités graphiques dans l'espace vide autour de leur position dans le but de supprimer la superposition (technique appelée *jittering* [156]), de celles (D) introduisant une déformation de l'espace conduite par l'intérêt de l'utilisateur (lentille *fisheye*, p. ex. [57]). Une différence remarquable entre les deux approches est que la première déplacera des entités graphiques se superposant à des positions différentes alors que la seconde non (C1).

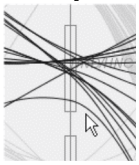
	ÉVITER LA SUPERPOSITION	ÊTRE SCALABLE	CONSERVER L'INFORMATION SPATIALE	MONTRER D'AUTRES ATTRIBUTS	MONTRER LA DENSITÉ	COORDONNÉES PARALLÈLES	NUAGE DE POINTS
	1.	2.	3.	4.	5.		
Altération de l'apparence des entités							
A. taille	◐		■	◐			[115, 169]
B. opacité	◐		■		■	[79]	[115]
Déformation spatiale							
C. déplacement vers un espace vide	■			■		[68, 23]	[94, 156]
D. déformation de l'espace (*fisheye*)	◐		◐	■		[57]	[24]
Abstraction							
E. échantillonnage	◐	■	■	■		[54]	[17, 33]
F. partitionnement	◐	■	◐	◐	◐	[114, 62]	[171]
G. carte de densité	■	■	◐		■	[79, 78]	[167, 31]

■ peut satisfaire • ◐ satisfait partiellement ou possiblement • □ non satisfait.

TABLE 2.4 – Techniques de réduction de l'encombrement visuel (d'après [52, tableau 3]) et exemples de techniques pour les nuages de points et coordonnées parallèles. Le tableau diffère du tableau de référence par l'ajout de G et en A4 car nous considérons que la diminution de la taille des marques est plus négative pour (4) que présenté par ELLIS et DIX [52].



(a) Avant déplacement.



(b) Après déplacement.

FIGURE 2.14: *Jittering* sur un axe des coordonnées parallèles [68].

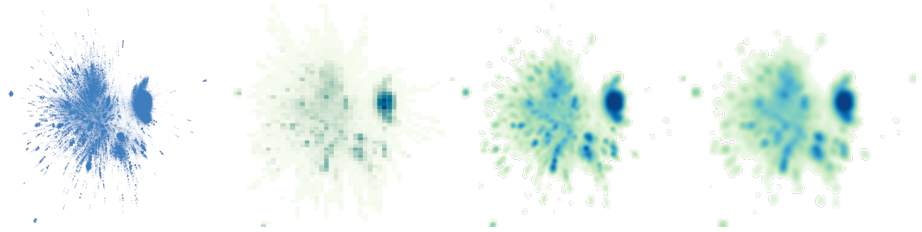
Dans la première catégorie (C), on trouve des techniques appliquées aux nuages de points [156, 94, 93] mais aussi aux coordonnées parallèles [68, 23] pour lesquelles un exemple est présenté en figure 2.14. Ces techniques font usage de l'espace vide pour réduire l'encombrement visuel et sont donc limitées à minima par la définition de l'écran (C2).

Dans la seconde catégorie (D), les techniques sont en général localisées et agrandissent localement une région de la vue déterminée interactivement par l'utilisateur [24, 57]. Ces déformations fournissent des indications de déformation qui permettent à l'utilisateur de conserver une connaissance du positionnement relatif des entités à la différence de celle de la première catégorie (C3, D3).

Abstraction

Les techniques d'échantillonnage (E) consistent à n'utiliser qu'un sous-ensemble des entités des données, assez représentatif. Par définition, la diminution du nombre d'entités a tendance à réduire la superposition. Dans un échantillon aléatoire, chaque entité des données a la même probabilité d'être sélectionné. En conséquence, l'échantillon peut ne pas être représentatif et ne pas retranscrire des structures importantes des données (p. ex. entités isolées). BERTINI et SANTUCCI [17] proposent une méthode d'échantillonnage aléatoire adaptatif pour les nuages de points destiné à préserver la densité *perçue*. La lentille d'échantillonnage [51, 53] utilise de l'échantillonnage aléatoire, dans les nuages de points et les coordonnées parallèles, sur une zone au diamètre et au taux d'échantillonnage paramétrables. L'échantillonnage permet d'accélérer les traitements en conservant le même idiome de visualisation. Cependant, une de ses limites est qu'il ne permet pas de tirer de conclusion sur les zones vides de la représentation.

Les approches par partitionnement (F) sont une autre forme d'abstraction réduisant le nombre d'entités graphiques par groupement des entités des données. Les entités graphiques représentent des ensembles d'entités et leur aspect visuel est conduit par des valeurs statistiques sur ces ensembles. Dans les coordonnées parallèles, des agrégats sont représentés par exemple par



(a) (b) (c) (d)

FIGURE 2.15 – Exemples de visualisations pour les mêmes données (≈*743 000* points) : (a) nuage de points conventionnel (opacité : 2 %), (b) carte de densité discrète par *binning* rectangulaire (≈*7000 bins*), (c) carte de densité lissée (KDE, noyau gaussien), (d) carte de densité lissée d'un échantillon des données (≈*7000* points).

une polyligne moyenne [62] ou par des surfaces délimitées par des courbes indiquant les extrema du groupe [114]. La superposition est réduite mais pas nécessairement éliminée (F1).

Les cartes de densité (G) représentent une fonction de la densité des points dans l'espace et sont parfois appelées histogrammes 2D (histogrammes lissés pour leur version lissée). La figure 2.15 en présente trois exemples. Sous leur forme discrète, elles sont un cas particulier d'agrégation issue d'un partitionnement *spatial* c.-à-d. partitionnement du domaine. L'agrégation par *binning* est l'exemple le plus commun : l'espace est divisé suivant un pavage par des rectangles ou des hexagones et les données résumées par le nombre d'entités présentes dans chaque cellule (ou *bin*) ainsi formé par une échelle de couleurs. BACHTHALER et WEISKOPF [5] proposent un modèle mathématique général pour des cartes lissées de nuages de points sous le nom de *continuous scatterplots* adapté ensuite aux coordonnées parallèles [78]. L'estimation à noyaux de la densité (KDE) est employée pour établir des cartes lisses de densité de points (p. ex. [102]) ou coordonnées parallèles (p. ex. [79]). L'approche *bin-summarize-smooth* [167] est un hybride des cartes discrètes et lissées qui fonctionne en deux étapes : une première de *binning* et une seconde de lissage, jugée importante pour réduire les variations excessives de valeur entre cellules adjacentes dues au procédé de *binning*. Comparé au partitionnement non-spatial, les cartes de densité éliminent par définition la superposition et montrent la densité des données (G1, G5). Leur efficacité est toutefois grandement dépendante de l'échelle de couleurs utilisée et ne permettent pas directement de montrer d'autres attributs des données (G4).

Les techniques d'abstraction sont conçues conjointement à des interactions de demande de détails qui permettent de changer interactivement la granularité de l'abstraction telle que décrites en section 2.1.2. Dans les contextes de l'échantillonnage (p. ex. [33]), du partitionnement (p. ex. [62, 55]) ou des cartes de densité (p. ex. [133]), ces interactions suivent parfois des approches hiérarchiques qui permettent d'obtenir des représentations *multi-échelle* réduisant les incohérences entre des vues de niveaux de détail successifs.

2.3.2 *Scalabilité computationnelle*

La principale difficulté du passage à l'échelle de système d'exploration visuelle de données est le traitement des données en temps interactif. Les techniques permettant la scalabilité computationnelle des traitements de données incluent des techniques matérielles ou architecturales, communes à d'autres domaines que la visualisation comme des systèmes de bases de données ou encore des paradigmes ou structures de données indépendants des idiomes utilisés. GODFREY et al. [66] classifient les nouveaux paradigmes de traitement de données, capables de passer à l'échelle, selon deux axes : l'utilisation de pré-traitement et la précision des résultats. Le premier axe correspond à la nature des données sur laquelle une requête ou un calcul est appliqué : soit brute, soit pré-traitée. Le second axe correspond à la nature des réponses attendues : soit exacte, soit approchée.

Pré-agrégation et indexation

L'abstraction hiérarchique [55] est une méthode de support de l'exploration multi-échelle de données. Elle consiste à pré-agréger les données à plusieurs niveaux de détails, les plus hauts utilisés comme support de la vue d'ensemble et les plus bas pour les vues détaillées. Cette approche est utilisée dans l'échantillonnage hiérarchique pour les nuages de points [33], le partitionnement par canopées hiérarchiques pour les cartes de densité lissées [133] et le partitionnement multi-dimensionnel pour les coordonnées parallèles [62].

Pour les interactions de brossage et lien sur des vues multiples, plusieurs travaux proposent des structures de données pré-calculées inspirées du principe des cubes de données. Un cube de données est un tableau multi-dimensionnel contenant les résultats de toutes les agrégations selon les différentes dimensions. La taille d'un cube de données croît exponentiellement avec le nombre de dimensions des données et leur *résolution* c.-à-d. nombre de valeurs différentes ou bien nombre de parties pour un partitionnement. imMens [107] et NanoCubes [105] sont des structures de données inspirées des cubes de données servant à accélérer des interactions de brossage et lien entre des vues agrégées. imMens utilise une structure de cube de données décomposée, appelées tuiles de données, qui permet de réduire l'empreinte mémoire et de stocker la structure complète dans la mémoire d'une machine mais ne supporte qu'une sélection à la fois. Falcon [119] propose une décomposition différente du cube de données, basée sur les vues initialisant l'interaction de brossage. Dans ces deux approches, chaque niveau de granularité requiert une structure de données. NanoCubes [105] et HashedCubes [100] sont des index hiérarchiques pour les données temporelles et spatiales qui organisent et agrègent les données selon chaque dimension. Rübel et al. [140] construisent un index utilisant les tableaux de bits (*bitmaps*) pour accélérer les interactions de surbrillance sur des coordonnées parallèles agrégées.

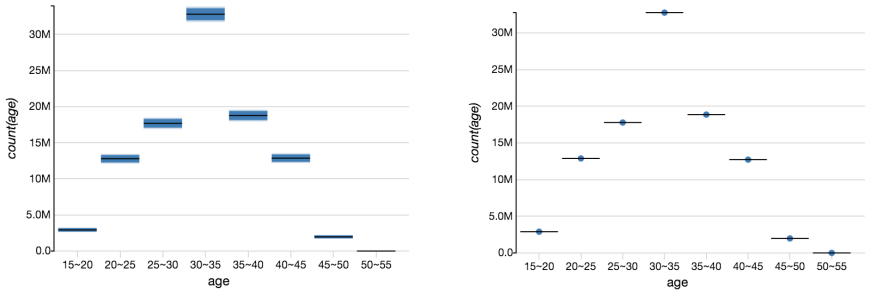
Parallélisation des traitements

La réduction des latences de traitement des données s'appuie aussi sur l'utilisation d'infrastructures dédiées de calcul et stockage. Rübel et al. [140] utilisent par exemple un super-calculateur pour accélérer les interactions de surbrillance dans des coordonnées parallèles agrégées. Les grappes de machines avec le système de stockage et calcul Hadoop et les outils de son écosystème (p. ex. HBase, Spark) sont utilisés pour le passage à l'échelle de la navigation dans des cartes de densités [133, 50] et des interactions de brossage et lien et demande de détail pour l'exploration de données multidimensionnelles [86]. Ces outils fonctionnant sur des infrastructures distribuées ont l'avantage d'être extensibles en ressources par l'ajout de machines *standards* à la grappe, de présenter une certaine tolérance aux pannes grâce à la réplication des données et d'être disponibles sur des environnements virtuels (*cloud*) sur lesquels l'augmentation des ressources est encore facilitée. Pour ces raisons leur coût initial est plus bas et leur accès plus facile que ceux des super-calculateurs.

Mise en cache et compression

Pour atténuer les latences dues au transfert par la liaison réseau dans les architectures client-serveur, certains systèmes cherchent à préparer les résultats à l'avance ou à minimiser l'empreinte du transfert par la compression. La préparation des résultats de requêtes d'interactions dépend de l'aptitude du système à anticiper les actions de l'utilisateur ou à mettre en cache les requêtes répétées. Doshi et al. [47] proposent un système adaptatif à plusieurs stratégies pour accommoder différents comportements utilisateurs et des variations au cours d'une même session d'exploration, appliqué notamment au brossage dans les coordonnées parallèles. ForeCache [8] s'intéresse aux interactions de navigation (zoom et déplacement) et à la prédiction des *tuiles* nécessaires aux prochaines interactions.

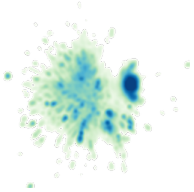
La réduction de la taille des données pré-calculées telles que stockées et transférées entre client et serveur est un autre axe de diminution des latences. Elmqvist et Fekete [55] présentent l'idée de *budget en entités visuelles* comme paramètre permettant de contrôler, voire d'assurer, un certaine



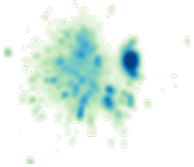
(a) (b)

FIGURE 2.17 – Exemple du paradigme progressif dans SwiftTuna [86] : un graphique converge de son état initial (a) à basse précision vers son état final (b).

réactivité de rendu en nombre d'images par seconde. Ce paramètre s'applique également au contrôle de la réactivité relative aux latences de transfert réseau pour les systèmes reposant sur un échange entre client et serveur à chaque interaction majeure. Ce principe est utilisé notamment pour la navigation dans les cartes de densité par PERROT et al. [133] : le nombre de points est borné à *1000* par tuile ce qui garantit des échanges de seulement quelques kilo-octets durant les interactions. Cette limitation du nombre de points a pour conséquence une diminution de la précision de la carte de densité et, en ce sens, cette approche est comparable à celle de la compression avec pertes [61]. La figure 2.16 montre un exemple d'une telle compression d'un facteur 1 :100.



(a) Carte de densité lissée (KDE, ≈*743 000* points).



(b) Carte de densité lissée approchée (aKDE, ≈*7000* points).

FIGURE 2.16: L'approximation de carte de densité lissée comme *compression* de la représentation (facteur 1 :100).

Approximation et paradigme progressif

L'*approximation* des résultats est un levier de réduction des latences d'interaction orthogonal au pré-calcul, au pré-requêtage et à la parallélisation des calculs. L'échantillonnage est un exemple générique d'approximation : un sous-ensemble des entités est utilisé pour estimer la valeur d'un calcul sur l'ensemble des entités ainsi que sa précision (p. ex. aKDE [133]). Cette idée est utilisée par le système BLINKDB [1], conjointement au pré-calcul d'échantillons, pour proposer à l'utilisateur (ou au concepteur d'un système l'utilisant) un budget pour une requête sous la forme de temps de calcul ou de précision du résultat. Ces techniques requièrent cependant des méthodes de visualisation spécifiques pour retranscrire l'incertitude induite par l'approximation et ce, notamment pour leur utilisation sous forme *incrémentale* ou *progressive* [86, 146, 153] où la représentation est progressivement raffinée d'une vue initiale à basse précision à une vue finale sans approximation comme illustré par le graphique de la figure 2.17.

3 CorFish : Mise en évidence scalable pour vues multiples

Dans ce chapitre, nous nous intéressons à la scalabilité visuelle d'interactions de brossage et lien pour des *vues multiples coordonnées* ou *liées* (cf. section 2.1.2, page 10). Les vues multiples sont des agencements juxtaposant plusieurs vues distinctes pour permettre l'analyse de données *complexes* en évitant le recours à une vue unique dont l'encodage serait probablement très complexe. Les vues sont distinctes par les données qu'elles représentent ou par les métaphores qu'elles utilisent, permettant ainsi d'analyser différents aspects des données. Par exemple, pour des données relationnelles dont les entités possèdent de multiples attributs, des vues différentes peuvent servir à représenter d'une part les relations entre entités, et d'autre part leurs attributs.

La contrepartie de la multiplicité des vues est tout d'abord la nécessité de recourir à l'interaction pour relier les vues entre elles et permettre à l'utilisateur d'analyser les données dans leur ensemble, et ensuite la réduction de l'espace écran disponible pour chaque vue. Pour réduire la probabilité de chevauchement entre entités visuelles, la taille de celles-ci doit donc être réduite ce qui peut diminuer l'effet de saillie induit par le changement de leur apparence comme le changement de couleur [150]. Ainsi, dans ce contexte, l'interaction de brossage et lien est à la fois essentielle mais aussi rendue moins efficace lorsque le nombre d'entités est grand. La problématique de ce chapitre est donc celle de la scalabilité visuelle des interactions de sélection et mise en évidence pour les vues multiples. Pour y répondre, nous avons proposé CorFish [1], une technique combinant une déformation spatiale à la méthode de surbrillance usuelle par la couleur.

ENCODAGE	DONNÉES PARTAGÉES		
	TOUT	SOUS-ENSEMBLE	AUCUNE
IDENTIQUE	*redondant*	overview+detail	
DIFFÉRENT	multiformes	overview+detail multiformes	*sans lien*

TABLE 3.1 – Possibilités de vues multiples selon le partage de données et d'encodage (d'après MUNZNER [121, p.276]). □ indique qu'une coordination par les entités est possible.

3.1 Problématique et existant

La coordination la plus commune dans des vues liées est celle de la sélection qui synchronise entre les vues la mise en surbrillance d'un sous-ensemble des entités. Cette interaction est généralement appelée brossage et lien, dénomination qui souligne les deux étapes qui la compose : tout d'abord le brossage qui définit un sous-ensemble d'entités sur une vue, puis l'immédiate mise en surbrillance faisant le lien entre les multiples représentations des entités de cette sélection (cf. figure 3.1). L'objectif de la surbrillance est de créer un effet de saillie (ou *emphase*) qui rende les entités sélectionnées proéminentes de sorte qu'elles puissent être identifiées en un instant [69].

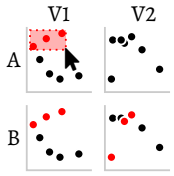


FIGURE 3.1: Brossage (A) et lien (B) sur deux vues liées (V1 et V2). Le lien fonctionne ici par la surbrillance (couleur rouge).

En général, l'efficacité de cet effet dépend de la manière dont le changement visuel opéré sur les entités sélectionnées s'intègre avec les autres encodages de la représentation mais aussi de la taille des entités. En effet, comme mentionné précédemment, plus la taille des entités visuelles est réduite, moins leurs changements d'encodage sont perceptibles. Dans ce contexte, la déformation spatiale peut être un moyen de remédier au manque d'espace écran en agrandissant certaines

1. pour **Co**ordinated **Fish**eye

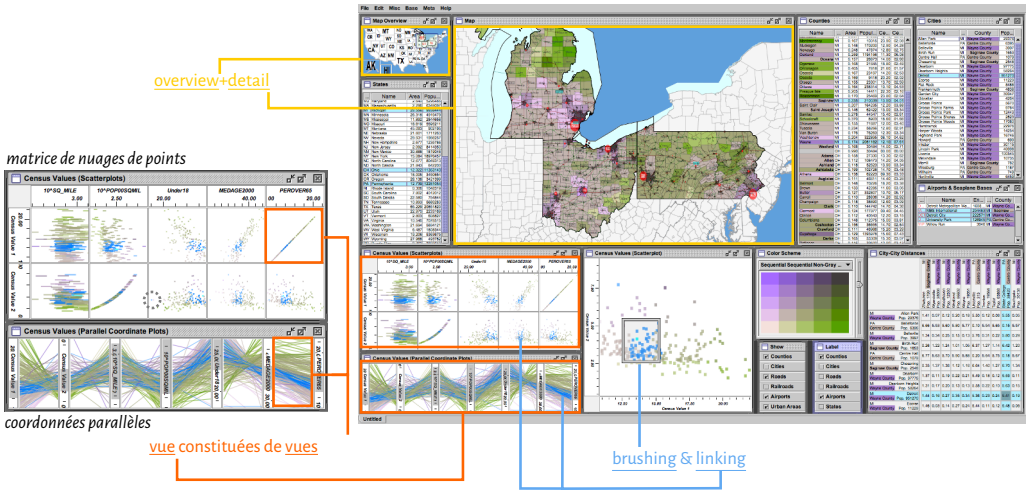


FIGURE 3.2 – Vue de l'outil Improvise [3] [166], présentant plusieurs vues, entre autres : une vue géographique en deux parties (vue d'ensemble et vue détail), des coordonnées parallèles et une matrice de nuages de points. Un ensemble d'entités, sélectionné depuis la vue nuage de points, est mis en surbrillance (en bleu) sur toutes les vues.

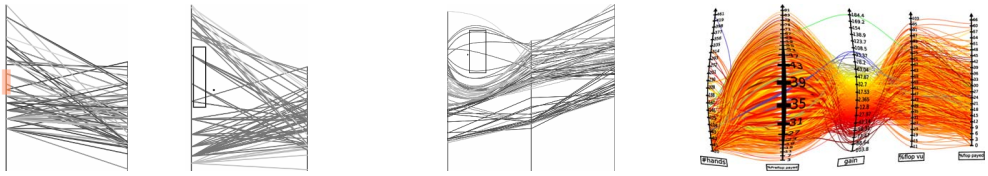
régions pour améliorer localement la visibilité d'entités et permettre de diriger l'attention de l'utilisateur sur ces zones élargies. Les principales difficultés de mise en place d'une déformation spatiale synchronisée sont la diversité des métaphores visuelles des différentes vues et la multiplicité des zones de déformation.

3.1.1 *Vues composées de vues*

On dit que des vues sont *distinctes* si elles renseignent sur différents aspects des données, par exemple en présentant une information différente ou en soulignant différents aspects de la même information [6]. Les techniques de visualisation pour données multi-dimensionnelles elles-mêmes peuvent être considérées comme constituées de multiples sous-vues correspondant à des représentations des sous-espaces des données. L'interaction de brossage et lien par exemple a été initialement conçue pour les matrices de nuages de points [9]. Les coordonnées parallèles et les matrices de nuages de points, sont des exemples de représentations décomposables en sous-vues, une par sous-espace 2D des données [2] qui sont les espaces séparant chaque paire d'axes pour les coordonnées parallèles et les éléments de la matrice pour les matrices de nuages de points. La figure 3.2 présente un exemple de vues liées, incluant une matrice de nuages de points et des coordonnées parallèles, où un ensemble d'entités, sélectionné par brossage sur un nuage de points, est mis en surbrillance en bleu sur les autres vues. La nature composée des matrices de nuages de points et des coordonnées parallèles les expose aux deux problématiques majeures des vues liées : (1) le maintien de *cohérence* [6, 135] dans l'encodage des vues, notamment au cours des interactions, et (2) l'encombrement visuel dû à la multiplicité des vues et à la quantité limitée d'espace écran.

Pour les coordonnées parallèles, de précédent travaux proposent des interactions de déformations spatiales pour atténuer l'encombrement visuel. Certaines interactions agrandissent localement une zone pointée par le curseur : sur les axes [57], entre les axes [57] ou sur l'ensemble de la vue (cf. figure 3.3, page 25). Sur ces exemples, la déformation est appliquée localement pour mettre en évidence un sous-ensemble des entités et n'est pas propagée aux autres sous-vues.

1. http://www.cs.ou.edu/~weaver/improvise/examples/census
2. Sous-espaces alignés aux dimensions du jeu de données.



(a) Déformation verticale sur un axe [57]. (b) Déformation radiale [57]. (c) Lentille *fisheye* dans l'espace image (extrait de [98]).

FIGURE 3.3 – Exemples de lentilles de déformation sur les coordonnées parallèles.

3.1.2 *Déformation spatiale et vues* fisheye

Les termes de *vues fisheye* et de *degré d'intérêt* (*doi* ou DOI) tirent leur origine des travaux de FURNAS [63] qui s'intéresse initialement au *filtrage dynamique* pour présenter, sur une même vue, une partie des données de manière détaillée et une vue d'ensemble du reste de manière moins détaillée. Dans sa définition, le degré d'intérêt d'un élément est une fonction $DOI(x, y) = API(x) - D(x, y)$ où API fournit un niveau d'intérêt intrinsèque, D est une fonction de distance, x la position d'un élément et y la position du point de *focus*. Le point de focus peut être unique (comme pour l'utilisation d'une lentille suivant le curseur de la souris) ou bien multiple. Cette fonction propose l'idée de *degré* d'intérêt, c'est-à-dire une définition de l'intérêt non plus binaire (sélectionné ou non), mais à plusieurs niveaux, voire par des valeurs dans un domaine continu. Cette idée est reprise dans la définition de HAUSER [70] ($DOI \in [0, 1]$) et dans le modèle en trois ensembles F, M et B de HALL et al. [69] où F et B désignent respectivement l'ensemble des éléments de plus grand et de plus faible intérêt alors que M regroupe des éléments ayant des niveaux d'intérêt intermédiaires.

Le terme de vues *fisheye* est aussi utilisé pour désigner des techniques utilisant la *déformation spatiale* plutôt que le filtrage dynamique pour intégrer une zone de focus dans une zone de contexte. De manière générale, la déformation spatiale peut pallier les deux problèmes liés : la restriction de l'espace écran en réduisant localement l'encombrement visuel, et la densité d'information en guidant l'attention de l'utilisateur vers une zone spécifique rendue saillante [92, 101]. Les déformations spatiales peuvent avoir des caractéristiques variées [29, 123]. Elles peuvent par exemple n'affecter que la position des éléments visuels (déplacement ou *déformation discrète* [123]) ou conjointement affecter leur position et leur forme (déplacement et agrandissement ou *déformation continue* [123]). Certaines méthodes de déformation sont générales et utilisent des fonctions de déformations 2D ou 1D [7] ou des manipulations de projection de vue 3D [28]. Les modèles EPF (Elastic Presentation Framework) [30] et *magnification fields* [91] définissent les déformations spatiales par des cartes de hauteur et le modèle *déformation-agrandissement* de LEUNG et APPERLEY [101] décrit plusieurs déformations 1D par des fonctions de déformation continues.

D'autres méthodes sont des interactions dédiées à une représentation spécifique comme la navigation dans les diagrammes nœuds-liens [99] et les tableaux [136], ou la réduction de l'encombrement visuel dans les nuages de points [93] et les coordonnées parallèles [125, 57]. Les techniques *multi-foci* agrandissent plusieurs zones à la fois pour les comparer [144, 56, 157]. Dans les cas où les zones de focus se chevauchent, KEAHEY et ROBERTSON [92] propose trois approches pour composer les déformations : l'application séquentielle, la segmentation et l'interpolation.

3.1.3 *Mise en évidence coordonnée*

Les interactions coordonnées entre plusieurs vues sont construites à partir d'une fonction de *couplage* spécifiant la correspondance d'entités entre vues, d'une fonction de *propagation* spécifiant les conditions de coordinations (sélection ou navigation [126]) et enfin d'une fonction de *rendu* dictant les changements visuels à propager [6]. KOYTEK et al. [97] décomposent la coordination en trois composants qui sont la *source* (ce qui a été sélectionné), le *lien* (l'expression de la relation)

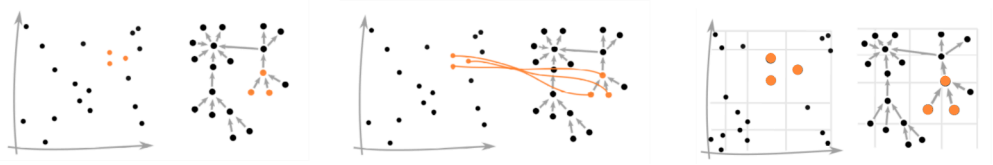


FIGURE 3.4 – Trois types d'interaction de sélection coordonnée : de gauche à droite, la surbrillance (ici par la teinte), la superposition (ici par des liens visuels), et la déformation spatiale.

(a) couleur (teinte) (b) netteté (c) taille (d) halo (e) opacité (f) forme

FIGURE 3.5 – Exemples de surbrillance (altération d'une variable visuelle non-spatiale) pour mettre en évidence une entité (d'après HALL et al. [69]).

FIGURE 3.6: Conflit de variable visuelle pendant la surbrillance. L'effet de mise en évidence de la surbrillance en rouge (à droite) peut être diminué par l'utilisation concurrente de la coloration pour encoder un attribut.

et la *cible* (les éléments qui sont en relation avec la source) et proposent une catégorisation des techniques de brossage et lien selon cette décomposition.

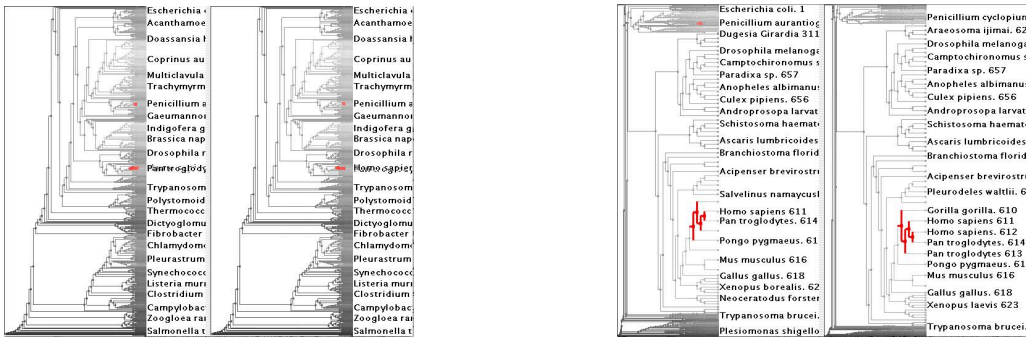
Une difficulté importante de la conception d'interactions coordonnées et générique est la difficulté à composer avec les différences entres les vues. D'une part, elles peuvent représenter différentes parties ou différentes granularités des données (entité ou agrégation d'entités) ce qui rend le couplage et donc la propagation complexe.

D'autre part, les différentes formes de vues rendent la dernière étape de rendu également complexe dû aux conflits potentiels d'usage de variable visuelle pour la sélection. Pour faciliter la compréhension des systèmes de vues liées, BALDONADO et al. [6] préconise de maintenir une uniformité dans les états et les interactions de toutes les vues. Ainsi ces vues doivent, autant que possible, présenter les mêmes potentialités d'interaction (cohérence des interactions) et propager les changements d'une vue à toutes les vues présentant les mêmes données (cohérence des états).

On distingue trois approches pour la mise en évidence d'entités dans des vues liées, résumées sur la figure 3.4 et dont les travaux correspondants sont décrits dans ce qui suit.

SURBRILLANCE La *surbrillance* est l'interaction la plus commune de brossage et lien. Elle consiste à encoder les valeurs d'intérêt avec une variable visuelle non-spatiale telle que la teinte [9, 46], la forme, l'opacité, etc (cf. figure 3.5). L'effet de mise en évidence peut être en conflit ou être rendu inefficace lorsque cette variable visuelle est aussi utilisée pour l'encodage d'un attribut des données, comme illustré sur la figure 3.6. La multiplicité des vues peut mener à une multiplication des variables visuelles utilisées et donc rendre difficile l'isolation d'une variable visuelle conjointement disponible sur toutes les vues.

SUPERPOSITION La *superposition* est la dernière catégorie et regroupe les techniques utilisant des éléments graphiques supplémentaires pour concentrer l'attention de l'utilisateur, apporter du détail ou faciliter l'identification des entités identiques sur les vues liées. Les liens visuels (*visual links* ou *leader lines*) en sont un premier exemple, illustré sur la figure 3.4, qui matérialise les connexions sémantiques entre entités visuelles du focus par des liens colorés. Pour limiter l'encombrement dû à ces liens, STEINBERGER et al. [151] utilisent un faisceautage (*bundling*) qui limitent le passage des liens sur les éléments saillants et sur les zones de couleur similaire. Les lentilles liées sont une seconde approche dont un exemple est l'extension de Bring&Go [120] proposée par DUBOIS et al. [49] qui synchronise des lentilles centrées sur une entité entre plusieurs diagrammes nœuds-liens. Les approches par superposition ont l'avantage d'être plus indépendantes des techniques de visualisation utilisées que les deux autres catégories (pour les liens visuels), et donc directement compatibles avec des vues multiformes. Elles sont toutefois



(a) Vue non déformée.

(b) Vue déformée.

FIGURE 3.7 – Comparaison de deux arbres phylogénétiques avec TreeJuxtaposer [122], les zones en surbrillance marquent les différences entre ces deux arbres et sont mises en avant par la déformation.

peu adaptées aux sélections de grands nombres d'entités puisque l'ajout d'éléments graphiques supplémentaires tend à créer de l'encombrement visuel.

DÉFORMATION SPATIALE Pour la représentation de données hiérarchiques, MUNZNER et al. [122] et GRAHAM et KENNEDY [67] ont montré l'efficacité de l'utilisation de techniques de *déformation spatiale* pour visualiser et comparer de grandes hiérarchies dont les entités seraient autrement représentées trop petites pour être manipulées ou distinguées. Les premiers utilisent une interaction de déformation appelée *AccordionDrawing* liant deux vues d'arbres phylogénétiques différents (cf. figure 3.7).

Des travaux existants ont exploré l'utilisation de la déformation spatiale interactive au sein de vues uniques pour différentes représentations ainsi qu'entre des représentations hiérarchiques liées, mais pas dans le contexte plus général de vues multiformes. Pourtant, comme la transparence ou la netteté, la déformation spatiale permet d'implémenter un filtrage visuel continu des éléments du contexte au profit de ceux du focus, ce qui en fait une technique potentiellement utile pour limiter le problème d'encombrement visuel des vues multiples et ainsi améliorer la scalabilité visuelle des interactions de mise en évidence d'entités.

3.1.4 *Propriétés attendues et comparaison*

Dans ce qui suit, nous définissons les propriétés attendues d'une interaction de mise en évidence pour qu'elle soit *scalable* et adaptée à un contexte de *vues multiples*.

F1 : SÉLECTION Pour être compatible avec plusieurs niveaux d'intérêt de l'utilisateur, assurer la composition d'opérations (raffinement) et être la plus générale possible, la définition de l'intérêt d'une entité doit être non-binaire (F1.1). Cette modélisation autorise ainsi la définition de sélections floues. Un second pré-requis pour la sélection est de pouvoir définir plusieurs entités d'intérêt à la fois (F1.2).

F2 : COORDINATION L'interaction doit être applicable à plusieurs vues à la fois (F2.1) et à plusieurs techniques de visualisation (F2.2) pour permettre de lier entre elles des vues de différentes formes [137].

F3 : SCALABILITÉ Le contexte des vues multiples présente des difficultés pour la mise en évidence d'entités dû à l'espace restreint de visualisation et au dessin de chaque entité visuelle sur peu de pixels [137]. L'effet de saillie doit être efficace pour un nombre croissant de vues (F3.1) et un nombre croissant d'entités (F3.2). De plus, l'interaction doit s'appliquer efficacement pour de petites et grandes sélections, c.-à-d. nombre d'entités dans le focus (F3.3). Ainsi, l'utilisateur peut raffiner une sélection initiale grossière en sélectionnant

	F1 : Sélection		F2 : Coordination		F3 : Scalabilité			F4 : Comparaison	
	Dégré d'intérêt non-binaire	Multi-entités/focus	Multi-vues	Multi-formes	Relativement au nombre de vues	Relativement au nombre d'entités	Relativement à la taille de la sélection	Avec le contexte	Entre sélections
	F1.1	F1.2	F2.1	F2.2	F3.1	F3.2	F3.3	F4.1	F4.2
Surbrillance									
1. Brushing & linking [9, 111]		■	■	▫			■	■	■
2. Smooth brushing [46]	■	■	■	▫			■	■	■
Superposition									
3. Liens explicites [40, 151]		■	■	■		■		■	◤
4. Lentilles [49]			■			■		■	
Déformation spatiale									
5. Vue simple [144, 157, 56]	■	■			■	■	▫		
6. Multi-vues [67, 122]	■	▫	■	■	▫	■	▫	◤	◤
7. CorFish	■	■	■	■	■	■	■	◤	◤

■ satisfait • ◤ satisfait en combinaison avec la teinte • ▫ partiellement satisfait • □ non satisfait explicitement.

TABLE 3.2: Comparaison de différentes approches pour la mise en évidence d'entités dans des vues liées, selon neuf critères.

progressivement des sous-ensembles de sa sélection initiale à mesure que la déformation spatiale le lui permet.

F4 : COMPARAISON Un des objectifs de la mise en évidence est de faciliter l'identification de groupe d'entités mais aussi de permettre leur séparation visuelle du reste (F4.1) [70], voire de permettre la comparaison de différents groupes d'entités (F4.2) [67, 151].

Ces propriétés sont utilisées pour comparer les techniques existantes de mise en évidence selon la catégorisation utilisée dans la section précédente, et pour évaluer CorFish et les résultats de ces comparaisons sont rapportés sur le tableau 3.2.

Déformation spatiale

La catégorie *déformation spatiale* réunit les travaux existants permettant plusieurs zones de focus puisque les entités d'intérêt ne sont pas nécessairement co-localisées. Ces méthodes permettent l'expression de plusieurs niveaux d'intérêt (F1.1) par la variation de l'intensité de l'agrandissement ce qui permet également de compenser la réduction de l'espace écran disponible pour chaque entité graphique lorsque le nombre de vues augmente (F3.1). L'agrandissement peut s'appliquer à un grand nombre d'entités d'intérêt puisque la compression des zones de contexte permet d'agrandir les zones d'intérêt, même lorsqu'elles ne sont pas visibles sur la vue initiale (F3.2). Pour les vues sans superposition entre les éléments graphiques comme les matrices, l'effet de mise en évidence est diminué lorsque de grands groupes d'entités sont sélectionnés [67] (F3.3). Les travaux existants s'intéressant à la coordination d'une déformation spatiale sont spécifiques aux représentations de hiérarchies de la même forme [122, 67]. Dans le cas général, la déformation spatiale uniquement ne permet pas l'identification des zones agrandies et donc des entités sélectionnées du reste (F4.1) ni la distinction de différentes sélections (F4.2).

Surbrillance

La surbrillance permet la définition de sélections multiples pour certaines variables visuelles comme la teinte ou la forme (F4.2) et permet en général la distinction entre contexte et sélection (F4.1). Ces méthodes deviennent moins efficaces, voire inutilisables, sans interaction de navigation, à mesure que les entités visuelles diminuent en taille (F3.1, F3.2) [150]. Dans certains cas, la variable de surbrillance interfère ou est en conflit avec l'encodage d'une vue et ainsi, sont effet peut être diminué (F2.2) [69].

Superposition

Par leur nature, les techniques de superposition sont potentiellement adaptables à de nombreuses formes de visualisation (F2.2) lorsqu'elles ne requièrent a priori aucune configuration particulière sur l'encodage des vues comme pour les liens visuels. Notamment, leur efficacité est peu dépendante de la taille des entités visuelles (F3.2). Les éléments graphiques supplémentaires permettent de distinguer sélection et contexte (F4.1) et leur coloration permet également de différencier plusieurs sélections (F4.2). Cependant, ces approches deviennent inutilisables lorsque chaque vue possède très peu d'espace écran (F3.1) et lorsque de nombreux éléments sont sélectionnés (F3.3). De plus, ces méthodes ne permettent pas directement des sélections non-binaires puisqu'elles reposent sur la présence ou absence d'éléments graphiques additionnels (F1.1). On pourrait cependant imaginer par exemple que l'opacité des liens visuels soit utilisée pour accomplir cet objectif.

CorFish a pour objectif de compléter ces méthodes en s'intéressant spécifiquement aux situations où les méthodes sans déformation spatiale sont peu efficaces dû au nombre de vues se partageant l'espace écran et au nombre d'entités représentées. La méthode proposée synchronise entre les vues des déformations spatiales orientées par les entités. À la différence de méthodes existantes utilisant la déformation spatiale, CorFish est conçue pour s'adapter facilement à de nombreuses formes de visualisation.

3.2 Déformation spatiale liée par les entités

On appelle *entités conceptuelles* l'ensemble $E = \{e_1, \ldots, e_n\}$ des entités des données à l'étude. Les entités conceptuelles possèdent des attributs et relations qui sont les supports d'un agencement de m vues juxtaposées qu'on notera $V_1, \ldots, V_m$. Pour simplifier le discours, nous considérerons que toutes les vues représentent tout le jeu de données, c.-à-d. que toutes les entités conceptuelles possèdent un élément graphique par vue. Dans le cas général, cette association n'est pas nécessairement ni surjective ni injective puisque certains éléments graphiques peuvent représenter des *groupes* d'entités ou encore aucune entité (cf. section 3.3). Cependant dans cette section, on s'intéressera au cas où cette association est bijective pour chaque vue, c.-à-d. qu'un élément graphique de chaque vue pourra être associé à une unique entité conceptuelle et on appellera cet élément graphique une *entité visuelle*. Cette association permet donc d'associer toutes les entités conceptuelles à une position sur chaque vue, celle de son entité visuelle.

Notre objectif est de synchroniser une déformation spatiale agrandissant les régions de chaque vue contenant les entités désignées d'intérêt à cet instant de l'exploration. On considère que l'intérêt d'une entité conceptuelle est définie par une fonction de degré d'intérêt, notée :

$$doi : E \to \mathbb{R}+,$$

et dont les valeurs sont redéfinies au cours des interactions. La fonction *doi* définit une notion de *focus* constitué de toutes les entités conceptuelles dont le *doi* est non-nul, et en conséquence de *contexte* constitué du reste des entités. L'objectif de l'interaction est donc que la déformation agrandisse les régions couvrant les entités visuelles des entités de plus haut intérêt.

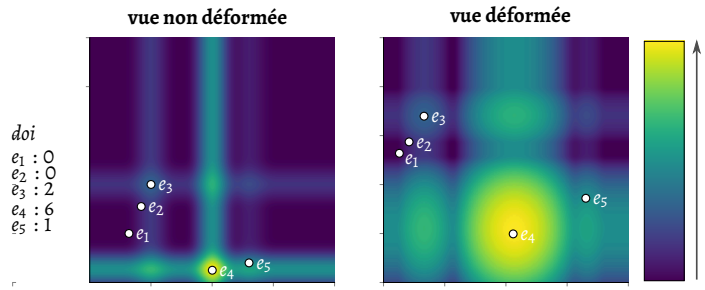


FIGURE 3.8 – D'une carte d'intérêt à une déformation de l'espace. À gauche, la carte d'intérêt, en couleurs, est calculée en moyennant les fonctions d'agrandissement définies sur les abscisses et ordonnées à partir des positions et du *doi* de cinq entités. À droite, les positions des entités et la carte sont déformées et l'on remarque l'expansion des zones d'intérêt (en jaune) et la compression de celles de moindre intérêt (en violet).

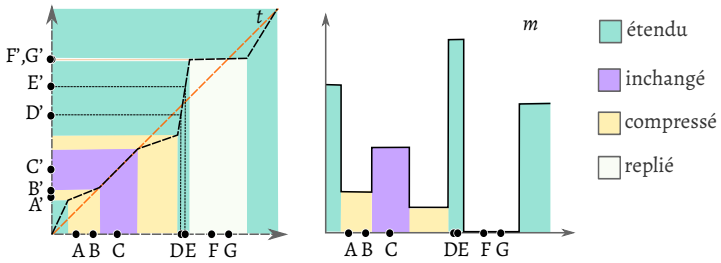


FIGURE 3.9 – Représentation d'une fonction de déformation t définissant une déformation d'un espace 1D sur lui-même (gauche) et de sa fonction d'agrandissement m (droite). Les variations de pente de t définissent différentes déformations du domaine de définition : certains intervalles sont compressés, d'autres étendus, inchangés ou repliés en un unique point.

3.2.1 *Propriétés attendues de la déformation*

La figure 3.8 illustre l'idée de traduction des degrés d'intérêt en déformation d'un espace continu. Sur la gauche, une échelle de couleurs représente les valeurs d'intérêt pour la vue, déduite des positions et *doi* de cinq entités. La carte d'intérêt est un champ scalaire 2D calculé en moyennant les champs scalaires des abscisses et des ordonnées, et qui a ses valeurs maximales autour de e_4. La déformation est calculée de sorte que ces valeurs d'intérêt correspondent à un facteur de déformation (expansion ou compression). Le résultat attendu est l'expansion des zones d'intérêt au détriment des autres, comme ce qu'on observe sur la figure 3.8 : sur le graphique déformé à droite, la zone jaune autour de e_4 est fortement élargie et les autres entités éloignées. Vu autrement, si l'intérêt est dilué par l'expansion et concentré par la compression, la déformation tend à le répartir uniformément.

De manière générale, une déformation d'un espace 1D est une fonction continue par morceaux de $\mathbb{R}$ dans $\mathbb{R}$. Suivant le formalisme de LEUNG et APPERLEY [101], cette fonction est appelée *déformation* et notée t. La déformation peut être définie de manière équivalente par la dérivée de t qui est appelée fonction d'*agrandissement* et notée m. La figure 3.9 représente un exemple de ce couple de fonctions pour une déformation ne modifiant pas l'échelle du domaine. En fonction des variations de t, certains intervalles du domaine de définition sont étendus ▫ par exemple $D'E' > DE$, d'autres compressés ▫ par exemple $A'B' < AB$, ou inchangés ▫ par la déformation. Un intervalle où l'agrandissement est nul ▫ sera replié sur un seul point comme F et G pour lesquels $F \neq G$ et $F' = G'$.

Soit une fonction $t : [0,1] \to [0,1]$, continue par morceaux sur $[0,1]$ et sa dérivée notée m. Cette fonction est une *fonction de déformation* pour un ensemble d'entités positionnés sur $[0,1]$ et leur fonction *doi* si et seulement si elle vérifie les propriétés suivantes :

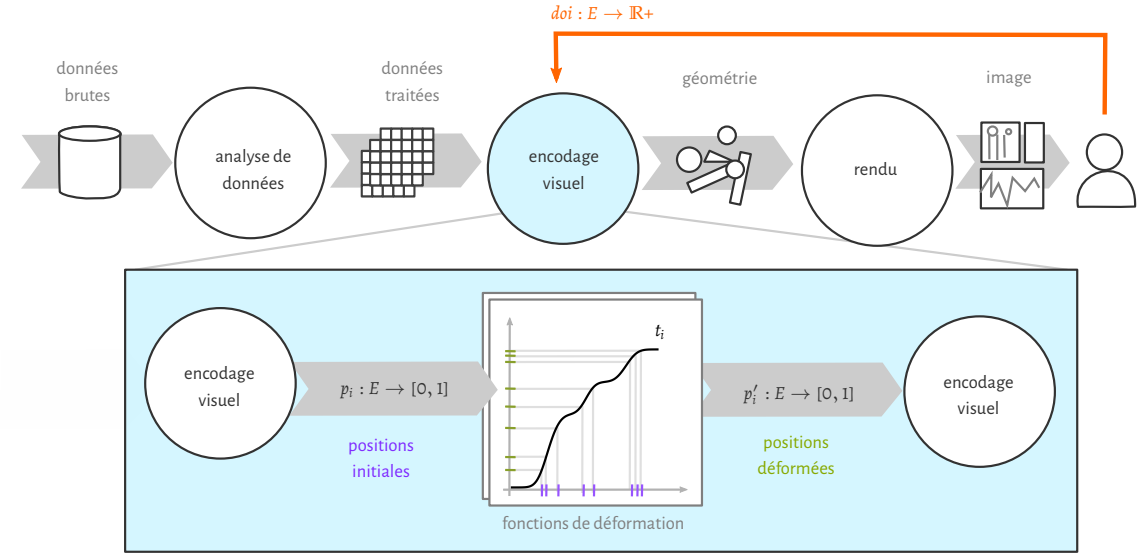


FIGURE 3.10: Le processus de déformation spatiale intervient à l'étape d'encodage visuel. Les fonctions de déformations t_i sont calculées pour chaque domaine 1D associé aux p_i à partir de la fonction *doi* courante. Ces fonctions de déformations sont ensuite appliquées aux positions initiales des entités visuelles pour obtenir les vues déformées : $p'_i = t_i \circ p_i$.

P1 : FOCUS t reflète les degrés d'intérêt de *doi* en étendant les intervalles contenant les entités de plus haut degré d'intérêt, m possède un maximum local autour des positions de ces entités.

P2 : ORDRE t préserve l'ordre des entités et donc est croissante, c.-à-d. m positive. Pour éviter de créer des superpositions d'éléments supplémentaires, m doit être strictement positive, c.-à-d. t doit être strictement croissante, autour des entités d'intérêt (cf. figure 3.9).

P3 : BORDURES Pour préserver l'espace de chaque vue après déformation, le domaine de définition et l'image de t doivent correspondre (l'image du domaine de définition de t par t est lui-même). Si t est croissante, cela revient à dire que $\int_{[0,1]} m = 1$.

3.2.2 *Pipeline : décomposition des vues en domaines 1D*

L'objectif de la méthode est de traduire l'intérêt de l'utilisateur défini sur l'ensemble *discret* des entités conceptuelles en des déformations des espaces *continus* de chaque vue. Ces déformations doivent agrandir les zones des vues contenant les entités de plus fort intérêt au détriment des autres et sont donc calculées à la fois à partir des positions initiales des entités visuelles et de leur *doi*. Pour tenir compte de la structure et de la sémantique des vues, les déformations ne sont pas appliquées dans l'espace image de chaque vue mais sur des domaines 1D qui correspondent aux domaines de coordonnées des vues. Comme illustré sur la figure 3.10, ce processus a lieu à l'étape d'encodage visuel du pipeline et utilise les positions initiales des entités visuelles.

Les vues 2D correspondent généralement à deux domaines de coordonnées (p. ex. nuage de points 2D), certaines vues à un seul (p. ex. une liste, matrice d'adjacence), et d'autres à plus (p. ex. coordonnées parallèles). Chaque entité visuelle d'une vue possède des coordonnées scalaires dans ces domaines de coordonnées. Des m vues, on extrait d fonctions de positionnement $(p_i)_{1 \leqslant i \leqslant d}$ où $p_i : E \to [0, 1]$ où les indices sont indépendants des vues. Ces fonctions de positionnement fournissent les positions des entités visuelles sur tous les domaines de coordonnées, sans redondance, destinés à recevoir une déformation. Dans une matrice de nuage de points par exemple, les abscisses des entités visuelles de chaque sous-vue d'une même colonne sont identiques et ne correspondraient donc qu'à une seule fonction de positionnement. Ce cas est discuté plus en détails dans la section 3.3. Déformer les vues revient à remplacer les fonctions p_i

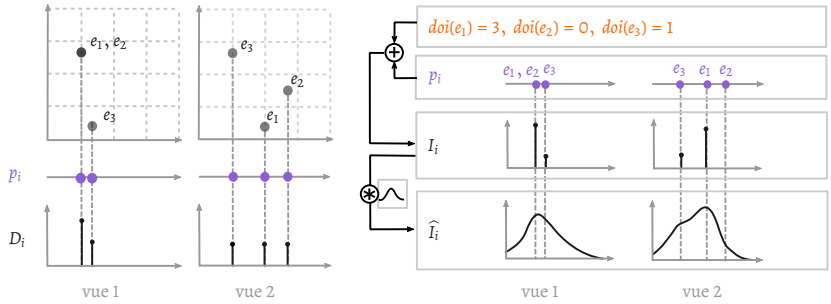


FIGURE 3.11 – Étapes de calcul de la fonction d'agrandissement $\widehat{I}_i$ pour les domaines des abscisses de deux nuages de points. À gauche : p_i pour les domaines des abscisses des deux vues et D_i fonction de distribution des valeurs de p_i. À droite : I_i fonction de distribution des valeurs de p_i pondérées par leur *doi* et $\widehat{I}_i$ lissage de I_i, pour une fonction noyau k_h.

par leurs composées $p'_i = t_i \circ p_i$ où chaque $t_i : [0,1] \to [0,1]$ correspond à la déformation calculée sur le domaine à partir du *doi*.

3.2.3 Fonction de déformation d'un espace 1D

Une fonction de déformation pour un domaine est calculée en termes de sa dérivée (fonction d'agrandissement), qui associe un degré d'intérêt à tout point du domaine. Pour que celle-ci reflète à la fois le *doi* et la distribution des entités, elle est définie dans le cas général comme une fonction lissée de distribution des entités où chaque position d'entités est pondérée par son degré d'intérêt. Ce processus est illustré pour deux domaines sur la figure 3.11 et détaillé dans ce qui suit. On supposera l'existence d'une entité focus, c.-à-d. qu'au moins une entité a un *doi* non-nul et ainsi que $\sum_{e \in E} doi(e) > 0$. Aucune déformation n'est à calculer dans le cas contraire.

Pour le i^e domaine, la fonction p_i donne les positions normalisées de toutes les entités et sa fonction de distribution $D_i : [0,1] \to \mathbb{N}$ associe à chaque point $x \in [0,1]$ le nombre d'entités positionnées en x :

$$\forall x \in [0,1],\ D_i(x) = \sum_{e \in E} \delta(x - p_i(e)) \text{ où } \delta(y) = \begin{cases} 1, & \text{si } y = 0 \\ 0, & \text{sinon} \end{cases} \tag{3.1}$$

Cette distribution est combinée au *doi* en pondérant chaque *impulsion* δ par la fonction $w : E \to [0,1]$ correspondant au *doi* normalisé comme illustré sur la figure 3.11, pour obtenir la fonction de distribution pondérée I_i représentant l'intérêt sur le domaine :

$$\forall x \in [0,1],\ I_i(x) = \sum_{e \in E} w(e) \cdot \delta(x - p_i(e)) \text{ avec } \forall e \in E,\ w(e) = \frac{doi(e)}{\sum_{f \in E} doi(f)} \tag{3.2}$$

Les poids de w sont les valeurs de *doi*, ajustées de sorte qu'elles se somment à un. Cette normalisation s'explique par le fait que dans un contexte de déformation spatiale sur un espace borné, les valeurs d'intérêt sont nécessairement relatives les unes aux autres plutôt que relative à une échelle commune. En effet, l'agrandissement de certaines régions correspond à une augmentation de l'espace écran qu'elles utilisent et donc nécessairement à une diminution de l'espace utilisé par d'autres, puisque la quantité totale d'espace écran reste la même. Une distribution pondérée I_i reflète l'intérêt de l'utilisateur sur un domaine mais sous cette forme, l'intégrale de cette fonction n'est pas une déformation valide. La distribution I_i est donc lissée pour obtenir $\widehat{I}_i$ comme illustré sur la figure 3.11. Le lissage résulte d'une convolution de I_i avec un noyau k_h :

$$\forall x \in [0,1],\ \widehat{I}_i(x) = (I_i * k_h)(x) = \int_{\mathbb{R}} I_i(x - y) \cdot k_h(y) \mathrm{d}y. \tag{3.3}$$

Les noyaux sont choisis parmi les fonctions intégrables, positives et symétriques qui admettent et atteignent leur maximum en 0, dont l'intégrale vaut 1 et dont le support, c.-à-d. la partie de son domaine où elle n'est pas nulle, est compact. On note k_h la contraction (horizontale et verticale) d'un facteur h du noyau k :

$$\forall x \in [0,1],\ k_h(x) = \frac{1}{h} \cdot k\left(\frac{x}{h}\right) \text{ avec } h > 0. \tag{3.4}$$

Le facteur h, appelé la *fenêtre*, contrôle l'étendue du noyau k_h et par conséquent la force du lissage produit par k_h sur $\widehat{I_i}$ (cf. section 3.2.4). Dans la suite on considérera que le support de k_h est $[-h/2, +h/2]$. Étant donnée la définition de I_i et la symétrie du noyau k, une autre expression pour $\widehat{I_i}$ est :

$$\widehat{I_i}(x) = \sum_{e \in E} w(e) \cdot k_h(x - p_i(e)) = \frac{1}{h} \sum_{e \in E} w(e) \cdot k\left(\frac{x - p_i(e)}{h}\right). \tag{3.5}$$

Le lissage est responsable de l'étalement de l'intérêt d'une entité conceptuelle autour des positions de chacune de ses entités visuelles dans un intervalle $h/2$, que nous nommons *aire d'influence* de l'entité visuelle. Notre définition des noyaux implique que toutes les entités focus ont la même taille d'aire d'influence, indépendamment de leur différence de degré d'intérêt. Plus précisément, chaque noyau non-nul est centré sur la position d'une entité focus par conséquent $\widehat{I_i}$ possède un maximum local dans l'aire d'influence de chaque entité pour l'entité e et le domaine i. En retour, la quantité d'expansion à un certain point du domaine dépend de sa proximité aux entités d'intérêt et de la taille des aires d'influences. Ainsi, $\widehat{I_i}$ vérifie la propriété P1 (*focus*) définie dans la section 3.2.1.

En tant que somme de fonctions positives, pondérées par des poids positifs, I_i et donc $\widehat{I_i}$ sont positives et strictement positives autour des entités focus ce qui vérifie la propriété P2 (*ordre*).

Pour vérifier la propriété P3 (*bordures*), l'intégrale de la fonction d'agrandissement doit valoir 1 sur $[0,1]$. Les entités proches des bordures du domaine, plus précisément à une distance inférieure à $h/2$, ne contribuent que partiellement à $\widehat{I_i}$ et alors $\int_{[0,1]} \widehat{I_i} < 1$ (comme le noyau centré sur e_1 sur la figure 3.12). Pour cette raison, nous définissons le terme de déformation de la fonction d'agrandissement m_i comme une normalisation de $\widehat{I_i}$. Pour moduler la force de la déformation, m_i est une interpolation linéaire de cette déformation avec la fonction d'agrandissement identité $x \mapsto 1$.

$$\forall x \in [0,1],\ m_i(x) = \alpha \cdot \underbrace{\frac{\widehat{I_i}(x)}{\int_0^1 \widehat{I_i}}}_{\substack{\text{terme de} \\ \text{déformation}}} + (1 - \alpha),\ \text{où } \alpha \in [0,1] \tag{3.6}$$

Ainsi, plus α est grand, plus fort est l'effet de déformation. De plus, en imposant $\alpha < 1$, m_i est strictement positive ce qui évite les situations de repliement. Sur la figure 3.12, par exemple, les intervalles grisés ne sont non-nuls pour m que grâce au fait que $\alpha < 1$ comme on peut le voir en comparant les courbes de m et $\widehat{I}$. Enfin, la fonction de déformation t_i correspondante est obtenue en intégrant m_i.

$$\forall x \in [0,1],\ t_i(x) = \int_0^x m_i(y)\mathrm{d}y = \alpha \cdot \underbrace{\int_0^x \frac{\widehat{I_i}(y)}{\int_0^1 \widehat{I_i}}\mathrm{d}y}_{\substack{\text{terme de} \\ \text{déformation}}} + (1 - \alpha) \cdot x \tag{3.7}$$

Trois exemples de fonctions d'agrandissement (m) et de leur fonction de déformation associée (t) sont représentées sur la figure 3.12 pour le même ensemble d'entités et différentes valeurs de *doi*. Le premier exemple est celui sans déformation ($\forall e \in E,\ doi(e) = 0$) : le terme de déformation est nul et donc la fonction d'agrandissement constante. En conséquence, la déformation est l'identité et les positions des entités sont inchangées. Sur le second exemple toutes les entités ont le même degré d'intérêt, non nul ($\forall e \in E,\ doi(e) = a > 0$). La quantité d'agrandissement

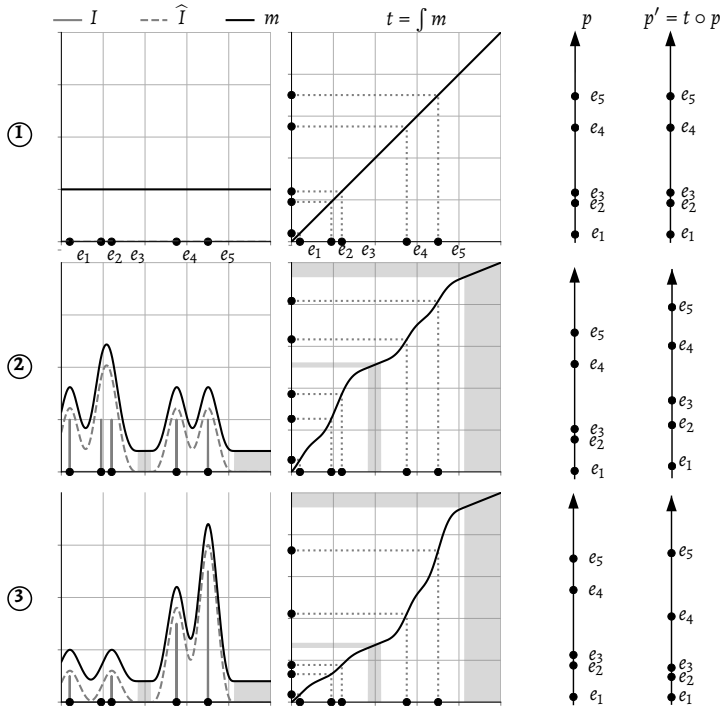


FIGURE 3.12 – Couples fonctions d'agrandissement (m) et de déformation (t) pour les entités conceptuelles $E = \{e_1, e_2, e_3, e_4, e_5\}$ dans trois cas de *doi* : ① *doi* nul pour toutes les entités, ② *doi* uniforme et ③ où e_4 et e_5 sont les entités de plus haut degré d'intérêt. Les intervalles grisés sont ceux des points de domaine hors de toutes les aires d'influences des entités ($\alpha = 0,6$).

sur les intervalles hors des aires d'influence (grisés) est nulle sur $\widehat{I}$ et dépend de α sur m. La fonction de déformation compresse ces intervalles et tend à répartir les entités uniformément sur le domaine. Sur le troisième exemple, e_4 et e_5 ont les plus fort degrés d'intérêt et sont séparées par la déformation. e_1 et e_3 ont un *doi* non nul mais relativement trop faible pour voir leur région agrandie.

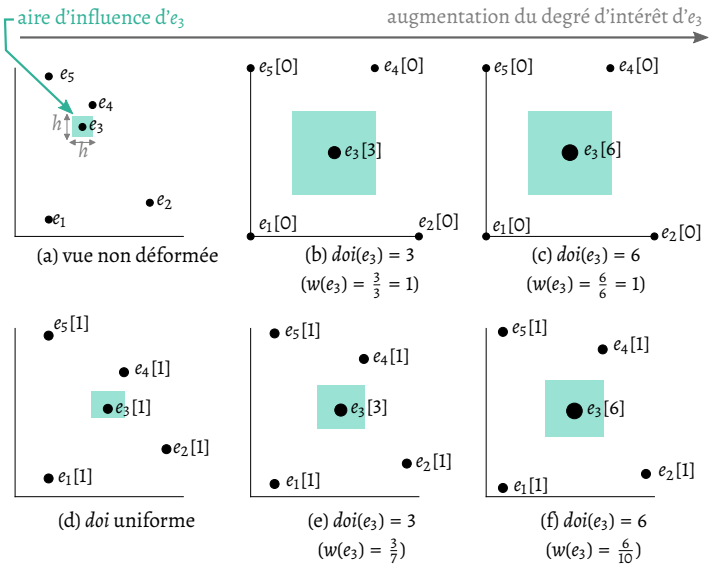
3.2.4 Ajustement de la déformation

Pour un domaine donné, l'allure de la déformation est affectée par la définition du degré d'intérêt mais aussi, secondairement, par les paramètres de lissage ainsi que le paramètre d'interpolation.

Degré d'intérêt doi

Une fonction *doi* non uniforme résulte en une fonction de déformation ré-allouant l'espace de manière à élargir les intervalles contenant les entités de plus fort intérêt *relatif*, au détriment des autres intervalles. Augmenter le degré d'intérêt d'une entité augmente le taux d'agrandissement de son aire d'influence et donc la rend plus séparée des autres entités. La figure 3.13 illustre cet effet pour l'entité e_3. Les vues de la première ligne montrent l'effet relatif du degré d'intérêt : les vues pour $doi(e_3) = 3$ et $doi(e_3) = 6$ sont identiques car correspondent à des degrés d'intérêt relatifs égaux.

Sur la seconde ligne, la première vue correspond à un *doi* uniforme : toutes les aires d'influence sont agrandies du même facteur de déformation ; les régions où se superposent les aires d'influence sont les plus agrandies. Après déformation les entités tendent à être plus uniformément positionnées dans l'espace. Les deux autres vues présentent des déformations calculées pour des *doi* non uniformes où e_3 a un degré d'importance plus grand que celui des autres entités. Ici,



(a) vue non déformée
(b) $doi(e_3) = 3$ ($w(e_3) = \frac{3}{3} = 1$)
(c) $doi(e_3) = 6$ ($w(e_3) = \frac{6}{6} = 1$)
(d) *doi* uniforme
(e) $doi(e_3) = 3$ ($w(e_3) = \frac{3}{7}$)
(f) $doi(e_3) = 6$ ($w(e_3) = \frac{6}{10}$)

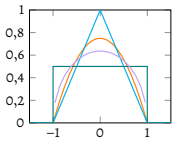
FIGURE 3.13 – Effet de l'augmentation du *doi* de l'entité e_3 relativement à des valeurs nulles de *doi* (a) et des valeurs uniformes de *doi* (d) pour les autres entités. (*doi* entre crochets, $\alpha = 1$, $h = 1/8$, noyau rectangulaire).

les déformations accordent plus d'importance à e_3 au détriment des zones vides et des zones incluant des entités au degré d'intérêt inférieur.

Fonction noyau k et fenêtre h

La forme de la fonction noyau k modèle l'allure de l'étalement de l'intérêt autour de chaque entité. Par exemple, la déformation induite par un noyau rectangulaire sera plus faible en son centre que celle induite par un noyau triangulaire (cf. figure 3.15). En plus de l'allure du noyau, la fenêtre h affecte la force du lissage. Une grande fenêtre résulte en une fonction d'agrandissement très lissée avec peu de variations et un effet de déformation faible ou absent. Au contraire, une petite fenêtre résulte en une fonction d'agrandissement plus contrastée qui correspond à un effet de déformation plus fort sur les centre des noyaux.

La figure 3.14 fournit un exemple de l'influence de h sur une déformation. La vue déformée de gauche présente la fenêtre la plus faible et la déformation la plus forte : e_3 et e_4, proches sur la vue de référence sont les plus éloignées sur cette vue. Au contraire, la déformation est la moins forte sur la vue de droite où toutes les entités sont proches de leur position de référence.



$T : x \mapsto (1 - |x|)$
$P : x \mapsto \frac{3}{4} \cdot (1 - x^2)$
$C : x \mapsto \frac{2}{\pi} \sqrt{1 - x^2}$
$R : x \mapsto \frac{1}{2}$

FIGURE 3.15: Exemples de noyaux au support $[-1, 1]$. T : triangulaire, P : parabolique, C : circulaire, R : rectangulaire.

Paramètre d'interpolation α

Ce paramètre contrôle le poids de la déformation spatiale dans son interpolation avec les positions initiales. Ainsi il compte simultanément pour l'intensité de la déformation et la compression des intervalles dénués d'entités, c.-à-d. les intervalles qui annulent la fonction d'agrandissement. La figure 3.16 présente deux exemples de déformations pour deux valeurs de α. L'effet de mise en évidence par la déformation spatiale a deux objectifs orthogonaux : produire un effet suffisamment fort pour efficacement mettre en évidence les entités d'intérêt et minimiser la déformation globale de la représentation de sorte que ses propriétés originales soient préservées. Le compromis optimal entre ces deux objectifs n'étant pas évident ou unique, il est utile de permettre de changer interactivement la valeur de α, par exemple via un curseur, pour passer progressivement d'un état de déformation à l'état original et inversement, et contrôler la force de la déformation.

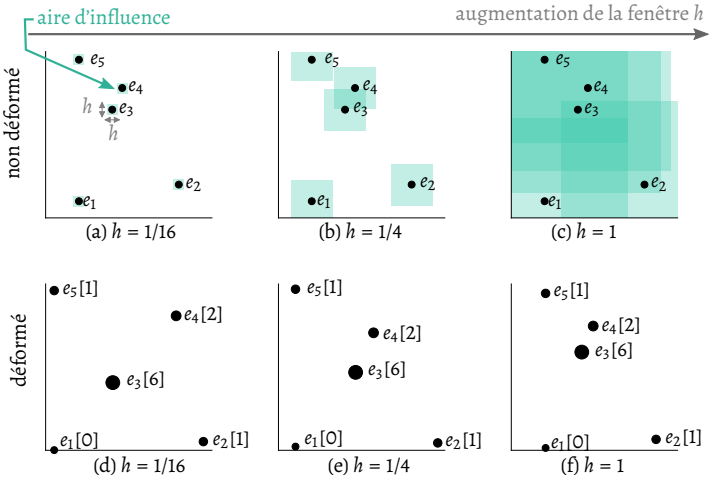


FIGURE 3.14 – Effet de l'augmentation de la fenêtre h. Plus la fenêtre augmente, plus la fonction de déformation est lisse et en conséquence moins l'effet de déformation est fort (*doi* entre crochets, $\alpha = 1$, noyau rectangulaire).

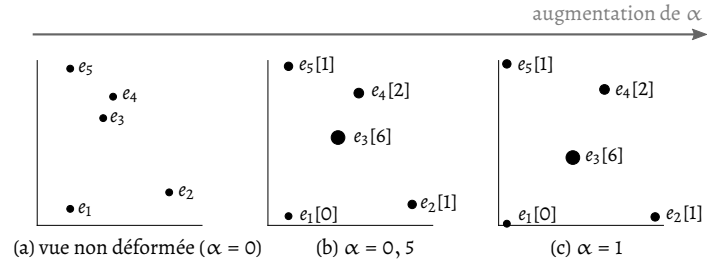


FIGURE 3.16 – Effet de l'augmentation du paramètre d'interpolation α (*doi* entre crochets, $h = 1/8$, noyau rectangulaire).

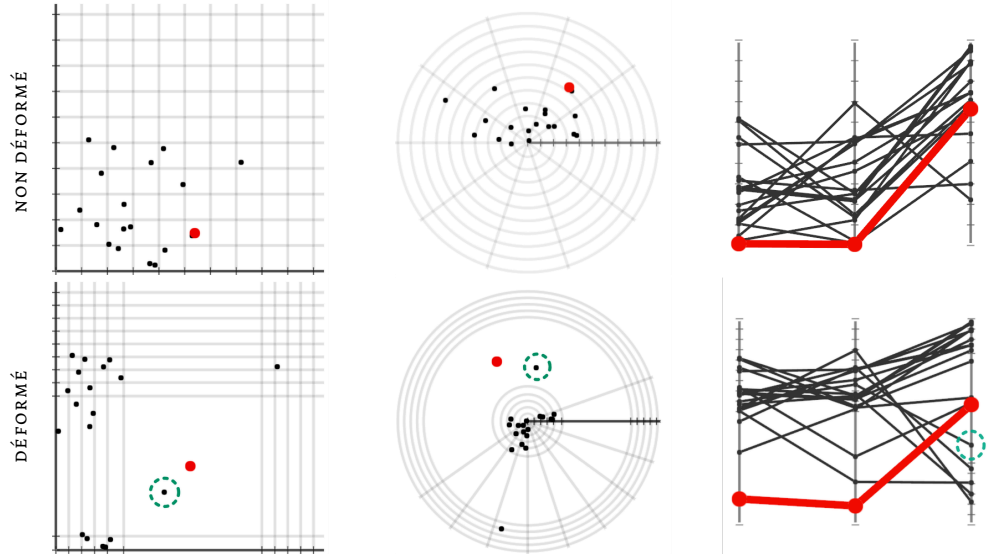
3.3 Application à différentes formes de visualisation

Dans les précédentes sections, nous avons présenté un système pour calculer une fonction de déformation d'un domaine 1D en fonction des positions et valeurs d'intérêt d'un ensemble entités. Dans cette section, nous présentons des cas d'application de ce système pour différentes formes de visualisation.

3.3.1 *Déplacement d'entités*

L'application de la méthode est directe pour les visualisations dont la position des entités visuelles dépend d'un repère de référence qui permet de définir un domaine de coordonnées et donc une fonction de déformation par axe. Sur un nuage de points par exemple, où les entités sont les points, les axes des abscisses et ordonnées forment deux domaines déformés séparément (cf. figure 3.17a). Sur des coordonnées parallèles où les entités sont les polylignes, chaque dimension ou axe constitue un domaine et les positions des sommets des polylignes sont déplacés (cf. figure 3.17c). La figure 3.17 montre un troisième exemple d'applications pour un nuage de points utilisant les deux domaines de coordonnées polaires. Un des atouts de la déformation par déplacement est de réduire localement l'encombrement visuel et la superposition. Cet effet est illustré sur la figure 3.17 par l'exhibition d'entités partiellement cachées, entourées sur les vues déformées où elles sont plus visibles.

Les diagrammes nœuds-liens dont les nœuds sont les entités sont un autre exemple de représentation où l'application est directe. Ici, les positions des entités ne sont pas immédiatement liées à un attribut des données mais résultent de l'exécution d'un algorithme de dessin. Dans



(a) Nuage de points (cartésien). (b) Nuage de points (polaire). (c) Coordonnées parallèles.

FIGURE 3.17 – Applications directes du déplacement des entités (doi(●)=1, doi(●)=0) sur une vue utilisant des coordonnées cartésiennes (a), polaires (b) et parallèles (c). Les entités encerclées sont peu visibles sur la vue non déformée mais clairement visibles sur la vue déformée.

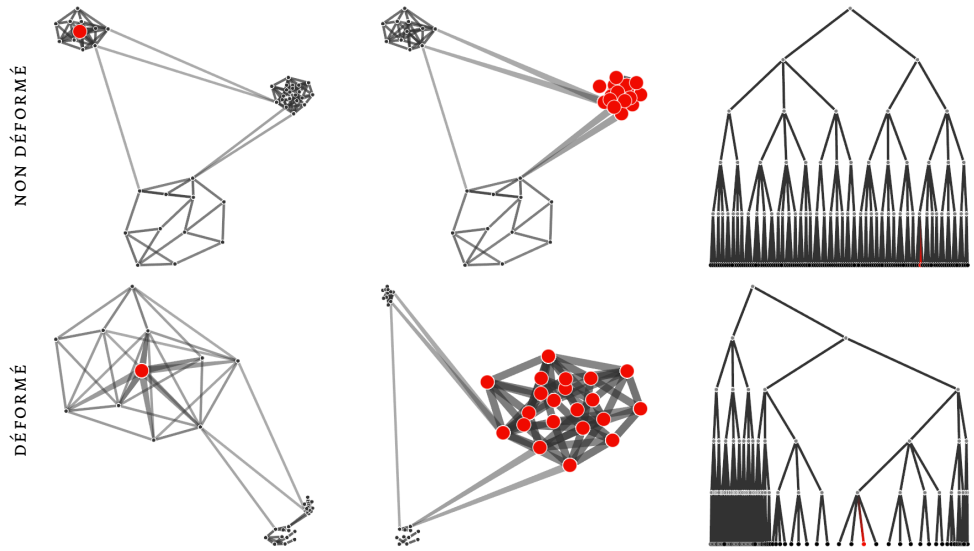
ce cas, notre approche considère les abscisses et ordonnées des positions résultant de ce dessin comme domaines de déformation. Les figure 3.18a montrent deux exemples de déformation sur un graphe à trois communautés. Sur la figure 3.18a, la déformation permet d'accéder au voisinage de l'entité focus puisque la totalité de la communauté à laquelle elle appartient est agrandie. Cela permet notamment d'identifier ses cinq voisins directs. Sur la figure 3.18b les entités focus appartiennent à la même communauté dont l'agrandissement permet d'en évaluer la taille (21 nœuds).

La figure 3.18c présente un exemple de dessin d'arbre dont les feuilles sont les entités. Ici, les entités sont alignées par le dessin, ainsi, seules les abscisses sont sélectionnées comme domaine de déformation. En plus d'être appliquée sur les positions des entités, la fonction de déformation résultante est aussi appliquée aux autres nœuds de l'arbre. Compte tenu des propriétés du dessin et de la préservation de l'ordre par la déformation, les nœuds parents restent verticalement alignés à leurs enfants, préservant ainsi la cohérence de la représentation. La déformation facilite ici l'examen et la sélection du voisinage de l'entité focus dont l'agrandissement provient de l'utilisation d'une fenêtre plus grande que l'espace entre les feuilles du dessin. Pour la mise en évidence de feuilles, cette approche résulte en une déformation similaire à celle de TreeJuxtaposer [122] lorsque la fenêtre ne correspond qu'à une unique feuille.

3.3.2 *Coordination de déplacement et agrandissement*

En fonction de la sémantique d'un domaine de coordonnées d'une vue et des formes des entités, la fonction de déformation peut être utilisée pour déformer la position des entités (déplacement), la taille des entités (agrandissement) ou encore les deux. Changer la taille des entités sans déplacement peut rapidement introduire de la superposition supplémentaire entre entités ce qui n'est pas souhaitable. Pour appliquer l'effet combiné de déplacement et d'agrandissement, nous déformons les points définissant les contours de chaque entité visuelle, dans nos exemples les deux points extrêmes délimitant la forme. La figure 3.19 présente trois exemples de cet effet combiné.

Pour la matrice d'adjacence de la figure 3.19a, chaque entité est associée à la fois à une ligne et une colonne de la matrice. Ici, puisque lignes et colonnes sont représentées dans le même



(a) Graphe (focus unique). (b) Graphe (multi-foci). (c) Arbre (focus unique).

FIGURE 3.18 – Application à des vues nœuds-liens d'un graphe et d'un arbre. Ici, les coordonnées cartésiennes sont utilisées comme domaine de déformation (seulement les abscisses pour c). (*doi*(●) = 1, *doi*(●) = 0).

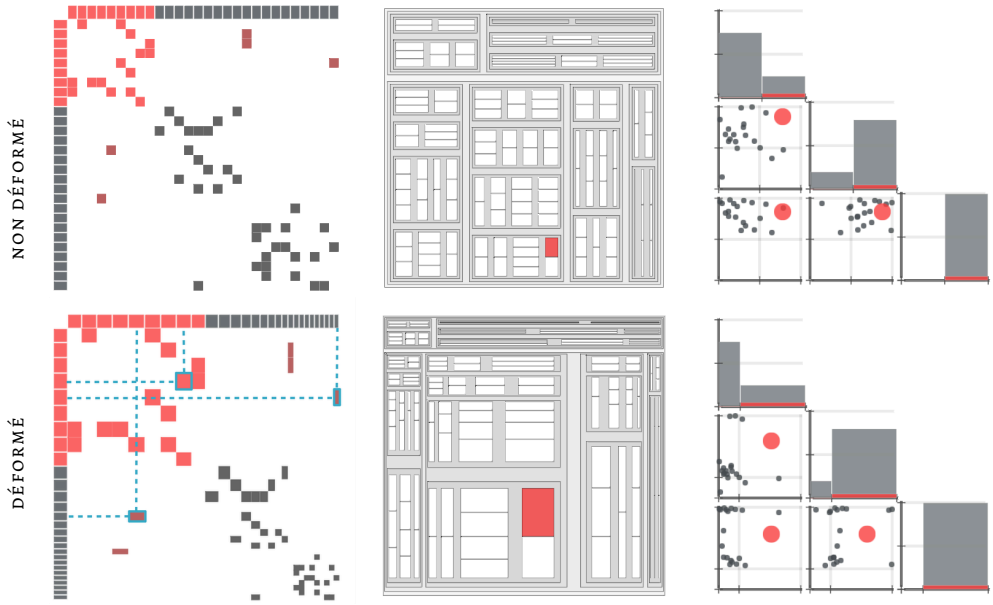
ordre, nous considérons un unique domaine de déformation dans lequel on associe à chaque entité deux positions correspondant à l'intervalle couvert par le rectangle de l'entité. On fait correspondre à chacune de ces positions la moitié du degré d'intérêt de l'entité correspondante. Après déformation de ces positions, chaque entité visuelle est dessinée de sorte qu'elle couvre les deux positions extrêmes déplacées. La fonction de déformation est aussi appliquée aux éléments graphiques représentant les arêtes, c.-à-d. les cellules de la matrice. Ainsi les arêtes connectant deux entités focus sont élargies dans les deux directions alors que les arêtes connectant une entité focus et une entité contexte sont élargies dans une direction seulement.

La figure 3.19b présente un exemple de *treemap* qui illustre le cas où les éléments déformés n'ont pas tous la même taille sur la vue initiale. Sur cet exemple, les entités correspondent aux feuilles de l'arbre représenté, c.-à-d. aux rectangles les plus intérieurs. On considère les abscisses et ordonnées comme domaines de déformations et les bordures de chaque rectangle comme positions des entités (2 par entité), traités comme celles de la matrice d'adjacence. Comme pour l'arbre de la figure 3.18c, les éléments graphiques correspondant aux nœuds internes de l'arbre sont déformés par les même fonctions de déformations qui déforment les entités. De nouveau, la préservation de l'ordre par la déformation assure que les propriétés d'inclusion restent valides dans la *treemap* déformée.

Dans les matrices de nuages de points, les histogrammes dessinés sur la diagonale correspondent chacun à une dimension des données (cf. figure 3.19c). Pour préserver la cohérence de la représentation, les positions extrêmes des barres des histogrammes sont déplacées par la même transformation que celle appliqués aux abscisses des nuages de points de leur colonne.

3.3.3 *Retour aux objectifs : combinaison à la surbrillance*

Nous présentons ici une évaluation de CORFISH au regard des objectifs énoncés en section 3.1, page 27. Comme présenté sur plusieurs exemples, CORFISH peut être appliqué à plusieurs formes de visualisation (F2.2). En tant que méthode de déformation spatiale, elle répond au critère de scalabilité pour les vues à faible résolution (F3.1 et F3.2). La déformation est pondérée par une fonction de degré d'intérêt non-binaire (F1.1). La déformation spatiale seule ne permet pas de distinguer plusieurs groupes d'entités (plusieurs sélections) ni de séparer clairement les entités



(a) Matrice d'adjacence. (b) *Treemap*. (c) Histogramme.

FIGURE 3.19 – Exemples de déformation conjointe de la taille et de la position (*doi*(●) = 1, *doi*(●) = 0).

focus des autres (F4.2 et F4.1). De plus, pour appréhender une vue déformée, l'utilisateur a besoin de signaux et indications permettant de déterminer qu'une déformation est à l'œuvre, ainsi que la manière dont elle opère pour éviter de l'induire en erreur. Ces deux derniers prérequis peuvent être remplis par l'ajout d'éléments visuels additionnels et d'un encodage visuel des sélections et de l'intérêt par la teinte et la taille.

Indications de déformation

Lorsque l'utilisateur connaît les positions des éléments graphiques pré-déformées, lorsqu'elles sont régulières comme par exemple sur une matrice, il peut évaluer la quantité et la localisation de la déformation. Dans les autres cas, la déformation à l'œuvre peut être transcrite par superposition d'une grille régulière déformée par la même transformation que les autres éléments graphiques de la vue l'ont été comme sur les exemples de la figure 3.17. Une autre possibilité consiste à représenter le champ scalaire de déformation par un gradient de couleurs faisant correspondre à chaque point de l'espace de chaque vue sa quantité d'agrandissement comme sur la figure 3.8, page 30. Cette approche peut se révéler plus précise qu'une grille mais a le désavantage de mobiliser au moins partiellement les variables visuelles de la couleur. Enfin, l'ajout de transitions animées entre les changements de *doi* et les changements interactifs de α et de la fenêtre h fournit une aide à la compréhension des déformations et à la préservation de la carte mentale de l'utilisateur.

Indications de l'intérêt et des sélections multiples

Distinguer les entités focus ($doi > 0$) des entités contexte ($doi = 0$) à partir d'une vue utilisant la déformation sans surbrillance est impossible dans le cas général car même si les transformations des vues sont déterminées par le *doi*, les valeurs du *doi* ne peuvent être déduite d'une vue déformée. Cela signifie que pour certaines vues, l'effet *extrinsèque* de mise en évidence par le déplacement spatial est en conflit avec l'effet *intrinsèque* de la distribution initiale des entités [69]. Pour remplir F4.1, les valeurs de *doi* peuvent être encodées sur la taille des entités visuelles en utilisant avantageusement l'espace créé par la déformation (cf. figures 3.17 et 3.18). La teinte des entités visuelles peut être utilisée pour encoder l'appartenance à la sélection ou une sélection dans le cas de sélections multiples (F4.2) comme sur la figure 3.20 qui présente un prototype

d'implémentation pour l'exploration d'un graphe de citations. Les entités visuelles d'intérêt sont agrandies par la déformation sur les vues s'appuyant sur le déplacement et l'agrandissement combiné (p. ex. figure 3.19), et sont agrandie par l'encodage des valeurs de *doi* sur la taille sur les vues n'utilisant que le déplacement (p. ex. figure 3.17).

3.3.4 Prototype d'implémentation

Pour mettre en application CorFish, nous avons implémenté un prototype d'application permettant d'explorer des données relationnelles dont les entités possèdent de nombreux attributs quantitatifs. La figure 3.20 page 43, présente un aperçu de l'application avec quatre des sept types de vues développés que sont la matrices de nuages de points et le nuages de points, les coordonnées parallèles, le diagramme nœuds-liens de graphes et d'arbre, la *treemap*, et la liste textuelle. Les fonctions de déformations ont été implémentées par des tables de correspondance (*lookup table*) dont la granularité est choisie en fonction de la fenêtre utilisée. Sur l'aperçu, deux sélections ont été définies pour un degré d'intérêt identiques et la taille et la teinte des marques encodent respectivement le degré d'intérêt et la sélection. Une fois une ou plusieurs sélections définies, leur visibilité relativement au contexte peut être ajustée interactivement à la molette par modification du paramètre d'interpolation α et de la fenêtre h.

3.4 Discussion

Dans un contexte de vues multiples, l'espace écran est partagé entre les vues donc limité pour chacune. La déformation spatiale est un mécanisme pertinent pour pallier cette limitation en ré-allouant l'espace écran en fonction de l'intérêt utilisateur et permettre la scalabilité de l'interaction usuelle de brossage et lien par la teinte. CorFish agrandit les régions en fonction du *doi* et des positions des entités ce qui permet, interactivement, de rendre plus visibles des éléments graphiques de taille initialement trop réduites pour qu'elles soient exploitables (p. ex. étiquettes) ou pour que leurs changements d'encodage soit perceptible (p. ex. couleur de remplissage).

Sur de nombreuses formes de visualisation, des techniques ad hoc de déformation spatiale sont préférable à CorFish puisqu'elles utilisent pleinement la structure des données d'origine et l'algorithme dont la représentation résulte (p. ex. utilisation de la topologie pour les diagrammes nœuds-liens [64]). En comparaison, le principal atout de CorFish est sa versatilité technique qui est avantageuse pour son application à de multiples formes de visualisation à la fois et donc au contexte de vues liées. Son implémentation commune pour différentes représentations peut aussi être un point de comparaison pour l'implémentation de techniques spécifiques à une forme de visualisation.

3.4.1 La déformation pour la scalabilité visuelle

La déformation spatiale sert ici à rendre un effet d'agrandissement qui montre plus de détails localement et peut aider à attirer l'attention de l'utilisateur sur certaines régions. Cependant la transformation des propriétés spatiales d'une visualisation peut rendre certaines tâches cognitivement plus coûteuses qu'avec de la surbrillance conventionnelle et surtout rendre certaines interprétations et raisonnement incorrects. De plus, comme mentionné dans la section 3.3.3, il est primordial que la présence de déformation soit identifiable pour éviter les interprétations erronées de la vue.

Pour la déformation par le déplacement uniquement dans les vues où la position encode des données quantitatives, tel que sur un nuage de points, la déformation n'est pas identifiable en général puisque la variable de position encode à la fois les données et le *doi* de manière non séparable. Pour permettre l'identification d'une déformation et faciliter le processus cognitif de lecture des données déformées, on ajoute donc des indicateurs de déformation comme les

grilles régulières déformées. L'ajout de graduations sur ces grilles permet d'estimer les valeurs originales des positions des entités sur une vue déformée, même si ce processus est évidement cognitivement plus coûteux que sur une vue non déformée et qu'il ne permet pas d'évaluer des distances.

Pour la déformation associant déplacement et agrandissement, nous distinguons deux cas. D'une part le cas des entités visuelles couvrant initialement des intervalles de taille identiques et sans chevauchement (p. ex. matrice d'adjacence) et d'autre part le cas des entités visuelles dont la taille encode une donnée (p. ex. *treemap* avec pondération). Dans le premier cas, l'effet d'agrandissement n'entre pas directement en conflit avec un encodage initial des aires des formes. Ainsi la déformation est identifiable directement, ce qui est l'effet escompté. Dans le second cas cependant, il ne semble pas pertinent d'appliquer notre méthode car une vue ainsi transformée ne préserverait pas assez d'information sur les données initialement encodées.

3.4.2 Approche 1D pour des vues 2D

Une limitation notable de CorFish est le calcul des fonctions de déformations sur des espaces 1D plutôt que dans le plan. Cette particularité rend l'approche directe mais limite son utilisation aux représentations possédant les symétries nécessaires pour être déformées sans changer dramatiquement de nature. Elle est aussi responsable de certains artefacts, notamment sur les vues nœuds-liens. Pour remédier à cette limitation, CorFish pourrait être étendue pour supporter les déformations 2D lorsque cela est opportun, par exemple en se basant sur des travaux existants [170, 175]. L'approche 1D a l'avantage de permettre la mutualisation des calculs de déformations pour les domaines identiques sur différentes vues ou sous-vues, pourvu que les déformations utilisent les mêmes paramètres. Sur une matrice de nuages de points par exemple, une seule fonction de déformation est calculée par dimension et réutilisée pour toutes les sous-vues de la même colonne et même lignes pour un *doi* donné.

3.4.3 Carte d'intérêt lissée et paramètres

Combinaison des déformations

Graham et Kennedy [67] a souligné la difficulté de la conception d'une déformation liée dans des nuages de points due à la possible superposition des entités visuelles qui n'existe pas pour les représentations de *treemap* ou matrices d'adjacences dans lesquelles les entités ne se chevauchent pas. Nous répondons à cette difficulté dans le cas du déplacement en combinant les noyaux pondérés se superposant par une somme. Par conséquence, sur chaque domaine, le facteur d'agrandissement en un point ne dépend pas uniquement du *doi* des entités du voisinage mais plutôt de l'accumulation de ces valeurs. Une limite de cette approche est donc qu'une forte densité d'entités au degré d'intérêt non-nul mais relativement faible résulte en une région dont le facteur d'agrandissement peut dépasser celui d'une région moins dense mais comportant une entité d'intérêt bien plus élevé. Ce comportement peut être inattendu mais aussi permettre de découvrir de telles régions d'entités d'intérêt co-localisées. Une étude utilisateur est nécessaire pour déterminer si d'autres combinaisons des noyaux (maximum, moyenne) sont préférables à l'usage.

Choix des paramètres

L'étape de lissage introduit une forme de complexité car elle nécessite de choisir plusieurs paramètres. Le choix de la fenêtre h détermine la force du lissage et donc l'étalement de l'intérêt de chaque entité. Plus la fenêtre est grande moins l'effet de déformation est fort. Le paramètre d'interpolation α influe aussi sur la force de la déformation en contrôlant la compression des régions dénuées d'entités ou ne contenant que des entités du contexte ($doi = 0$). Ainsi, nous supposons que pouvoir changer ces paramètres interactivement contribue grandement à la compréhension

de la déformation à l'œuvre, au maintien de la carte mentale de l'utilisateur et à l'utilisabilité de l'interaction. Dans l'objectif de simplifier le paramétrage, ces deux paramètres pourraient être abstraits en un seul méta-paramètre qui servirait à l'utilisateur pour contrôler interactivement la force de déformation. Une autre possibilité serait de déterminer automatiquement la fenêtre h exhibant un bon taux de lissage pour chaque domaine, en fonction de la distribution des positions. Le paramètre α resterait tout de même sous le contrôle de l'utilisateur puisqu'il permet de passer progressivement de la vue déformée à la vue non-déformée et inversement. Enfin, le noyau est le paramètre ayant le moins d'importance lorsqu'il y a beaucoup d'entités, ainsi il peut être choisi sur la base d'arguments purement computationnels par exemple (noyau rectangulaire ou parabolique).

3.5 Conclusion

Dans ce chapitre, nous avons présenté une technique de coordination de déformation spatiale, semblable à un effet *fisheye*, pour la mise en évidence d'entités dans les vues multiples et multiformes. L'utilisation d'une déformation spatiale est motivée par le fait que les approches conventionnelles de surbrillance perdent en efficacité lorsque les entités visuelles ne sont représentées que sur peu de pixels. Notre objectif est de parvenir à maintenir l'efficacité de la surbrillance lorsque le nombre d'entités visualisées grandit au-delà du million d'entités.

Notre approche utilise une définition commune et non-binaire de l'intérêt sur les entités qui modélise l'intérêt de l'utilisateur tel que défini par un brossage par exemple. À partir de ces valeurs, les vues sont déformées de sorte que la quantité d'agrandissement d'une région traduise l'intérêt des entités s'y trouvant. Les positions des entités sur chaque vue sont déformées en combinant des déformations des domaines de coordonnées des vues. En combinant l'agrandissement des régions d'intérêt à des encodages conventionnelles de la surbrillance par la teinte et la taille notre méthode remplit les besoins identifiés pour le passage à l'échelle d'interactions de brossage et lien entre vues multiples et multiformes.

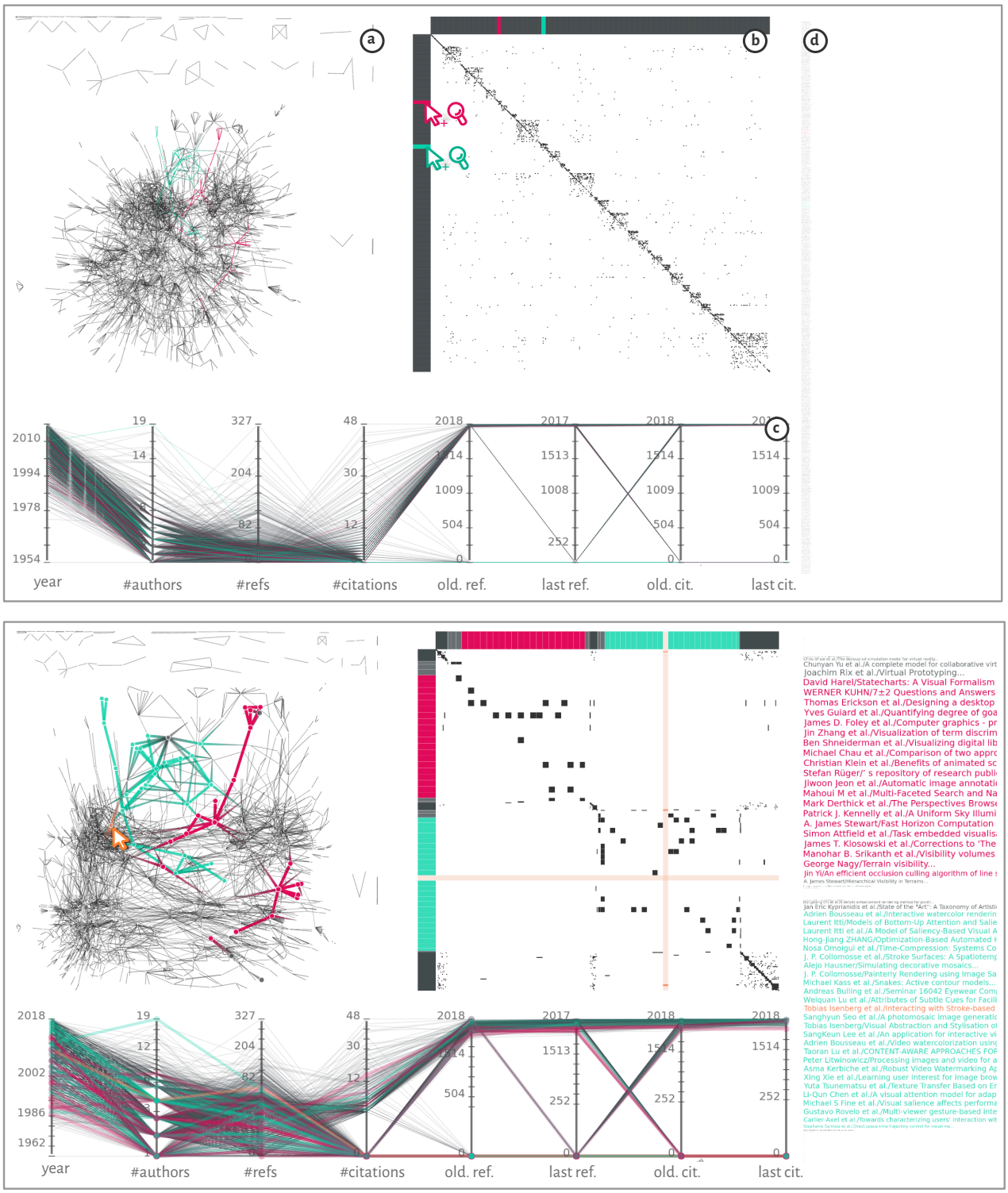


FIGURE 3.20: Application dans un prototype d'implémentation utilisant quatre vues pour analyser des données relationnelles (graphe de citations entre articles de recherches) dont les entités possèdent huit attributs quantitatifs et un nom (le titre accompagné de son premier auteur). Les relations entre entités sont représentées sur un diagramme nœuds-liens (a) et une matrice d'adjacence (b). Les attributs quantitatifs sont représentées sur des coordonnées parallèles (c) et les noms des entités sur une liste (d). En haut, deux sélections sont définies par le biais de la matrice d'adjacences. En bas, les entités correspondant aux deux sélections sont mise en évidence simultanément sur toutes les vues et une de ces entités est mise en surbrillance (en orange) par survol de son nœud sur la vue nœuds-liens.

4 CoReDeM : Simplification géométrique pour les cartes de densité

Dans ce chapitre, nous nous intéressons à la scalabilité computationnelle de la visualisation interactive de nuages de points représentés sous forme de cartes de densité. Dans un contexte de données massives, le système de visualisation est divisé entre une infrastructure distante et un client de visualisation déporté (cf. chapitre 2, page 11). La problématique de ce chapitre est la reconstruction de cartes de densité à partir de données géométriques réduites pour permettre l'exploration de grands nuages de points à bas coût de transfert et de stockage des différents échelles pré-calculées. Ce chapitre présente CoReDeM [1], une approche générale de compression géométrique servant à reconstruire une carte de densité à partir de données réduites, ainsi qu'une comparaison de plusieurs stratégies l'implémentant.

4.1 Contexte

Le nuage de points (*scatterplot*) est une représentation commune pour visualiser des ensembles d'entités à deux ou trois attributs quantitatifs. Il permet d'effectuer plusieurs tâches comme évaluer la corrélation entre les attributs ou encore analyser la distribution des points en identifiant des groupes de points et des points aberrants. La représentation conventionnelle dessine une marque [15] pour chaque entité. De ce fait, l'accroissement du nombre d'entités et leur concentration sur certaines régions d'une vue tendent à provoquer de la superposition ce qui résulte en de la perte d'information (cf. section 2.3.1).

4.1.1 *La carte de densité comme passage à l'échelle visuel*

La démocratisation des périphériques géolocalisés et des plateformes collaboratives de cartographies font des données géographiques une source de grands jeux de données, de plus de quelques millions de points, dont certains exemples utilisés dans la littérature sont présentés sur le tableau 4.1 comme les enregistrements géolocalisés du réseau social Brightkite ou encore les départs et destinations des courses de taxis à New York. Ces données sont aussi parfois dynamiques, comme les 2,7 milliards de points d'OpenStreetMap qui subissent 3 millions de changements par jour [2].

NOM	TAILLE
Brightkite	4
Higgs	11
OAG	200
US Census	300
TLC NYC Taxi	1100
OSM	2700

TABLE 4.1: Exemples de grands ensembles de points (taille en millions). Brightkite, OSM et TLC NYC Taxi sont des données géographiques (localisations).

La visualisation de grands ensembles de points posent tout d'abord un problème de passage à l'échelle visuelle : l'accumulation de points provoque de l'encombrement visuel. L'altération de la taille et de l'opacité des marques ou le *jittering* sont efficaces à petite échelle mais rapidement limitées [133, 52] comme décrit dans le chapitre 2, page 13. Les approches par cartes de densité consistent à transformer l'ensemble de points en un champ scalaire de densité qui est représenté par une échelle de couleurs comme l'illustre les exemples de la figure 4.1. Cette approche présente donc des données agrégées, de manière semblable à l'agrégation implicite sous-jacente à la composition des marques semi-transparentes. Cette agrégation peut théoriquement induire une perte d'information puisque le nombre de niveaux de densité associables à des couleurs différentes est limité par les caractéristiques matérielles des écrans ($256^3 \simeq 170$ milliards de couleurs). En pratique, cette perte d'information est avant tout due aux limites perceptuelles humaines et à la difficulté de concevoir de *bonnes* échelles de couleurs et une fonction de correspondance entre valeurs de densité et couleur qui soit adaptée à la distribution des valeurs de densité [16]. Si elles sont quantifiées, les limites de perception peuvent même être exploitées pour améliorer

1. pour **Co**nservative **Re**duction of **De**nsity **M**aps
2. https://wiki.openstreetmap.org/wiki/Stats

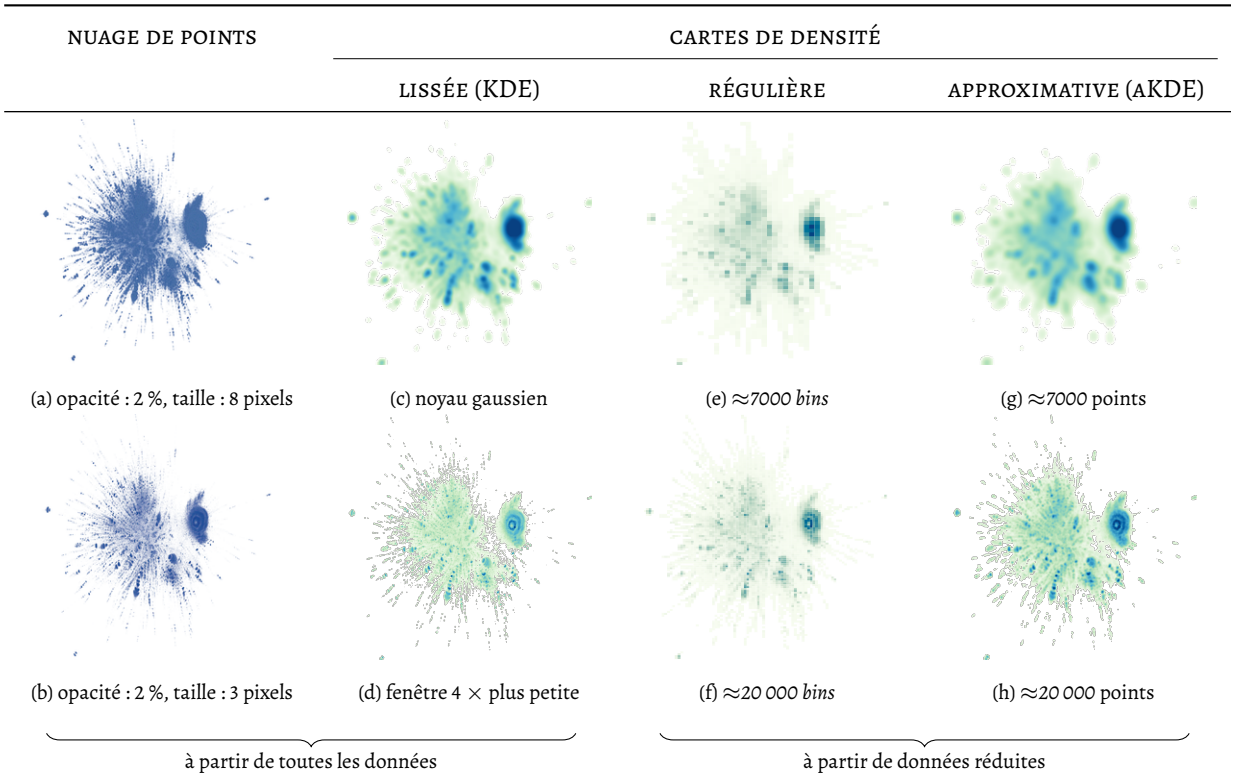


FIGURE 4.1: Différentes formes de nuages de points et cartes de densités pour un ensemble d'environ *743 000* points sur des images 256 × 256 pixels. À gauche : représentations basées sur l'ensemble des données, à droite : celles qui peuvent être reconstruites à partir de données réduites (pré-agrégées).

l'interactivité en produisant une image de qualité *juste suffisante* pour ne comporter aucune différence perceptible à l'image de référence [66].

De manière générale, la visualisation de grands ensembles de points induit donc une perte d'information due à la fois aux limites de ressources (définition et gamut de l'écran) et aux limites du système visuel et cognitif humain qui traite ensuite l'image. Dans ce contexte, les interactions de navigation (zoom et déplacement) sont donc essentielles pour permettre d'accéder aux détails.

4.1.2 *Pyramide d'abstraction pour la visualisation déportée interactive*

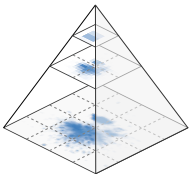


FIGURE 4.3: Pyramide d'images pour la navigation multi-échelle. Chaque niveau est subdivisé en tuiles.

L'exploration de grands ensembles de nuages de points introduit une difficulté supplémentaire liée aux latences induites par les temps de chargement, transfert, traitement et rendu des données. Lorsque les données brutes ne tiennent pas dans la mémoire d'une machine standard, des infrastructures spécifiques, généralement distantes, sont nécessaires. Cela implique donc des échanges via une liaison réseau entre le client de visualisation et ces plateformes au cours des interactions. Différentes stratégies de division du pipeline de visualisation sont décrites en chapitre 2, page 11. Nous reprenons ici deux stratégies adaptées aux données massives.

La première (S1 sur la figure 4.2) consiste à transférer les données de visualisation sous la forme d'images [34]. Dans cette approche la bande passante utilisée est directement liée à la définition des images et donc en pratique bornée. Des images de différentes définitions peuvent être pré-calculées en pyramide pour permettre la navigation multi-échelle dans le jeu de données en réduisant ou éliminant ainsi les latences de traitement pendant l'interaction (cf. figure 4.3). Une pyramide d'images est constituée de plusieurs images du même jeu de données, une par niveau, produites à des définitions croissantes du niveau le plus haut au plus bas. Chaque niveau est subdivisé en sous-parties de définition égale, appelées tuiles. Les interactions de zoom et

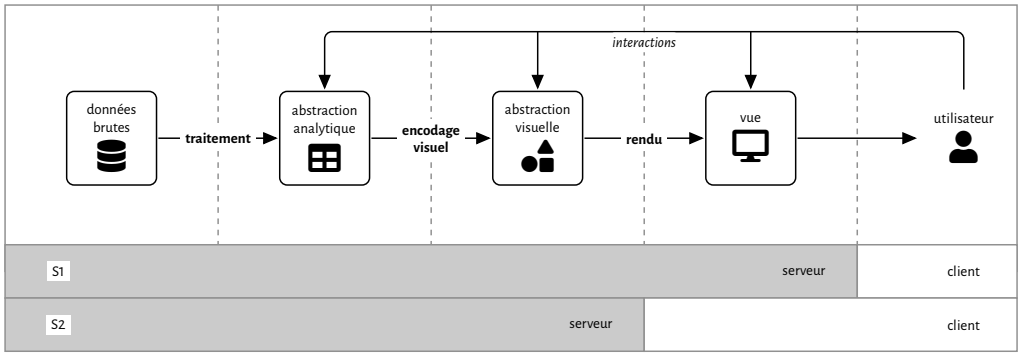
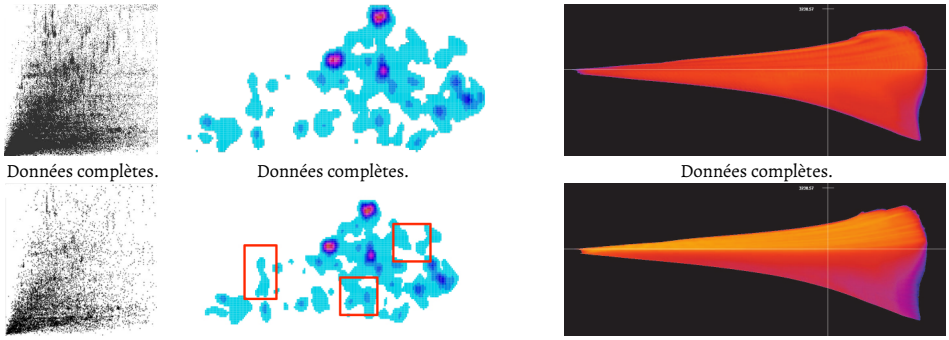


FIGURE 4.2 – Deux stratégies de division du pipeline de visualisation déportée. La première stratégie (S1) découpe le pipeline après l'étape de rendu, et transfère sur le réseau des données image. La seconde stratégie (S2) découpe le pipeline après l'étape d'encodage visuel, et transfère des données géométriques, laissant l'étape de rendu au client.

déplacement sont en général conçues de sorte qu'une vue couvre un nombre borné de tuiles. Ainsi une action de déplacement ne nécessite le transfert que de quelques tuiles du même niveau, et une action de zoom de quelques tuiles d'un niveau supérieur ou inférieur. Le mécanisme de pyramide d'images a un coût de stockage exponentiel : pour une pyramide de 10 niveaux avec des tuiles de taille 256×256, les images de la pyramide représentent 90 milliards de pixels soit plus de *2 To* non compressés et plus de *2 Eo* pour 20 niveaux. Ce coût nécessite donc des infrastructures largement dimensionnées de sorte qu'elles puissent également mettre à jour cette structure lors de changement dans le jeu de données, et notamment être capable de suivre le rythme de mise à jour. D'autre part, bien qu'atténué par la compression, le stockage sous forme d'images des nuages de points peut comporter un fort taux de redondance lié aux nombreuses zones de vide. En effet, à la différence de données photographiques par exemple, les ensembles de points réels sont des données *discrètes* par nature. Enfin, dans un contexte de visualisation, le stockage et l'échange d'images limitent fortement les possibilités d'interactivité pour le client de visualisation. La reconfiguration d'une échelle de couleurs, le redimensionnement d'une fenêtre ainsi que les interactions de détail au survol et de coordination de sélection entre vues par exemple ne sont pas directement possibles sur le client.

La seconde stratégie (S2 sur la figure 4.2) consiste à transférer des informations géométriques plutôt que des images. PERROT et al. [133] par exemple transfèrent au client un sous-ensemble de points pondérés utilisé ensuite pour reconstruire, du côté client, une carte de densité lissée (cf. figure 4.1d). Cette carte est une approximation de la carte de densité utilisant les mêmes paramètres de lissage mais l'ensemble des points du jeu de données plutôt qu'un ensemble réduit de représentants. De manière similaire à la pyramide d'images, ils construisent un arbre de partitionnement analogue à la pyramide d'images qui sert à borner le nombre de points transférés au client pour tout état de zoom et de déplacement. Pour un jeu de données de $\approx$*636 000* points différents et 21 niveaux, l'espace de stockage occupé par la pyramide de points n'est que de *380 Mo* [133]. Pour assurer le lien entre les vues, LIU et al. [107] stockent également des tuiles de données plutôt que des tuiles image, où les données correspondent à des projections multi-dimensionnelles. Leur approche s'appuie sur une agrégation par *binning* servant à rendre des cartes de densité semblables à celle de la figure 4.1c. Ces deux approches consistent à pré-calculer et transférer des données *réduites* ou *agrégées* qui sont utilisées ensuite par le client pour reconstituer la vue. Le transfert de données est plus flexible que le transfert d'images, car il découpe le pipeline de visualisation plus en amont, ce qui laisse une plus grande part d'interactivité potentielle au client de visualisation. De plus, une structure multi-échelle de données/géométrie a une empreinte de stockage plus faible que celle d'une pyramide d'images pour les données discrètes.



Données complètes. Données complètes. Données complètes.

Échantillon (22 %). Échantillon non-aléatoire (≈ 4 %). Échantillon uniforme à basse-résolution.

(a) À partir d'une métrique [17]. (b) Carte de densité à partir d'un *coreset* [177]. (c) Nuages de points *continus* [5].

FIGURE 4.4 – Exemples de techniques d'échantillonnage pour reconstruire une vue du nuage de points ou carte de densité à partir d'un échantillon des points.

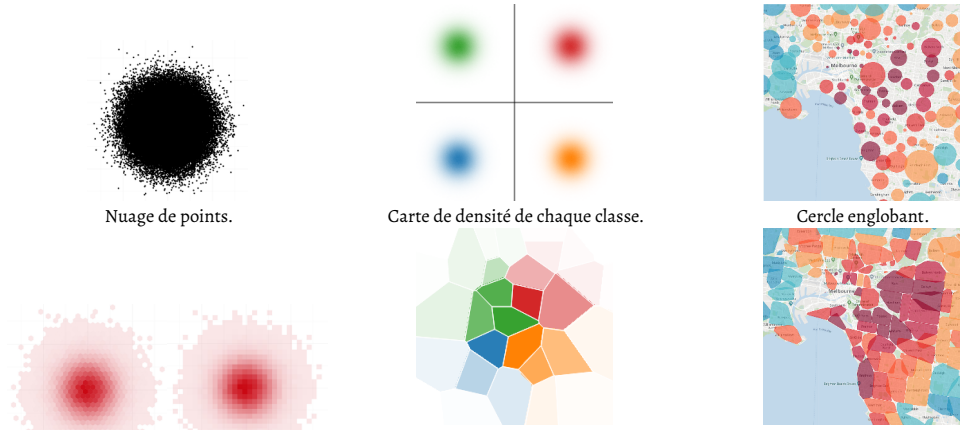
4.2 Cartes de densité à partir de données réduites

On recense dans cette section les techniques permettant la reconstruction d'une carte de densité ou d'un nuage de points à partir de données de plus petite taille que les données d'origine, c.-à-d. des données *réduites*. La visualisation ainsi reconstruite implique nécessairement une perte d'information à partir d'un certain taux de réduction des données. Parmi les travaux présentés ici, certains ont pour objectif de réduire l'encombrement visuel relativement à une représentation conventionnelle de nuage de points plutôt que celui de reconstruire une représentation à partir de données réduites.

4.2.1 *Échantillonnage*

Une première manière de réduire l'ensemble de points est de sélectionner un sous-ensemble des points pour la représentation. Cette technique cherche à conserver dans l'échantillon une densité relative apparente semblable à celle des données complètes. Si l'échantillonnage est fort, alors les zones des données de faible densité risque d'apparaître vides. De plus, les différences de densités ne sont pas perçues de la même manière à toutes les échelles [17]. L'étude expérimentale menée par BERTINI et SANTUCCI [17] a montré que les différences de densité de points entre deux zones sont décelées le plus souvent lorsque au moins une des deux zones a une densité supérieure à 60 %. Ils proposent donc deux processus d'échantillonnage destinés à préserver les caractéristiques de l'image de référence. Le *meilleur* échantillonnage aléatoire, illustré sur la figure 4.4a, est le résultat d'une recherche du taux d'échantillonnage minimisant la métrique évaluant les différences de densité entre régions non perçues par l'utilisateur. CHEN et al. [33] combinent une technique d'abstraction visuelle consistant à échantillonner la fonction de densité des points des données à une approche hiérarchique qui permet d'éviter les incohérences entre plusieurs niveaux de zoom.

Certaines méthodes utilisent un échantillon de points pour construire une carte de densité lissée. ZHENG et al. [176] construisent un échantillon du jeu de données, appelé *coreset*, permettant d'approcher la fonction d'estimation de la densité (KDE) de l'ensemble des points avec une certaine probabilité et une certaine précision, toutes deux paramètres du processus. Cette approche permet de reconstruire une carte de densité lissée très proche de celle de référence (cf. figure 4.4b). PERROT et al. [133] utilisent un partitionnement par canopées pour élire un échantillon pondéré de points et calculer une estimation pondérée de la densité dont la ressemblance à la carte exacte a été expérimentalement montré par calcul de score de similarité [164]. BACHTHALER et WEISKOPF [5] proposent une technique reconstruisant un champ scalaire de densité à partir d'un échantillon. Comme pour l'estimation par noyaux, la carte de densité résultante est continue ce qui a l'avantage



Nuage de points.

Carte de densité de chaque classe.

Cercle englobant.

Enveloppe concave.

(a) Binning rectangulaire et hexagonal (d'après [107]).

(b) Carte de densité combinée par diagramme Voronoï aléatoire pour des données multi-classes [87].

(c) Deux représentations de ConcaveCubes [103].

FIGURE 4.5 – Exemple de représentations agrégées de carte de densité.

de lisser les données et combler des zones vides. Leur technique obtient de bons résultats pour des échantillons de taille faible (cf. figure 4.4c). Ces techniques proposent des manières de compenser, au moins partiellement, le désavantage principal de l'échantillonnage qui est ne pas pouvoir tirer de conclusion concernant les zones de faible densité de la représentation puisque par définition, certains points sont écartés. Leur désavantage est qu'en retour elles « étalent » les valeurs de densité à des régions ne comportant pas de points dans les données originales.

4.2.2 *Agrégation*

Le *binning*, illustré sur la figure 4.5a, est l'exemple canonique d'agrégation pour les nuages de points. Il consiste à partitionner l'espace selon un pavage régulier et faire correspondre à chaque cellule ainsi constituée le nombre de points des données leur appartenant. La valeur de chaque cellule est ensuite encodée par une couleur [31, 167, 16]. Cette approche est efficace en termes de calcul et la régularité de sa représentation présente des avantages pour la comparaison de la densité entre plusieurs zones. Trois pavages réguliers du plan sont possibles, triangles, rectangles et hexagones, mais seuls les deux derniers sont en général considérés notamment car ils permettent à toutes les cellules d'être orientées à l'identique. Le *binning* hexagonal est celui qui résulte en la plus faible erreur moyenne entre un point et le centre de sa cellule [147]. Les pavages irréguliers sous la forme de diagrammes de Voronoï issus d'ensembles de points aléatoires ont également été éprouvés pour agréger des données [87] (cf. figure 4.5b). Un diagramme de Voronoï est un pavage non-régulier du plan en cellules à partir d'un ensemble de points, appelées germes, de sorte que chaque cellule ne contienne qu'une seule germe et qu'elle forme l'ensemble des points du plan les plus proches de cette germe que d'aucune germe.

Une limite des partitionnements spatiaux comme le *binning* est leur indépendance de la structure des données qui peut contribuer au masquage de structures fines dans les données. Au contraire, l'agrégation des points peut suivre un partitionnement orienté par les données [103, 171]. Dans les matrices de nuages de points hiérarchiques [171], les points sont partitionnés selon chaque dimension et chaque partie est représentée par un rectangle aligné avec les axes qui couvre la région occupée par les points de la partie. Pour des données géographiques, ConcaveCubes [103] partitionne les points avec DBScan et représente les parties par des enveloppes concaves non chevauchantes (cf. figure 4.5c). Pour pré-calculer des interactions de brossage et lien, ces représentations agrégées sont aussi appelées cartes de densité discrètes et couplées à des structures inspirées des cubes de données [100, 105, 103]. (cf. chapitre 2, page 20).

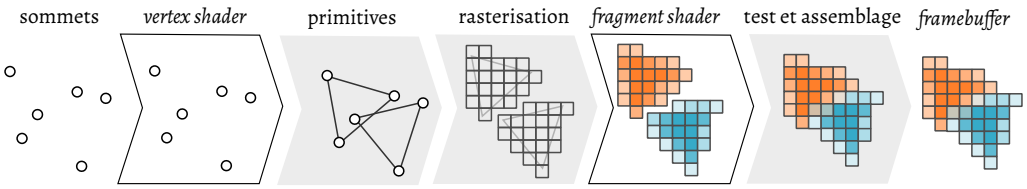


FIGURE 4.6 – Pipeline de rendu graphique pour deux primitives.

Les fortes variations entre agrégats voisins peuvent distraire l'utilisateur des tendances générales ce qui constitue une seconde limite des cartes de densité issues de partitionnements [167]. Pour atténuer ce phénomène, plusieurs travaux utilisent un rendu lissé des données agrégées, par exemple en appliquant un lissage par noyau gaussien sur un *binning* [167] ou l'utilisation de gradient de couleur centré sur un point moyenne au sein de chaque groupe [171]. D'autres travaux appliquent une agrégation au niveau pixel comme *Information Mural* [85], NANOCUBES [105] ou les cartes de densité de BERTINI et al. [16].

Dans ce travail, nous avons proposé d'explorer l'efficacité de cartes de densité non lissées, reconstruites à partir de formes discrètes (p. ex. rectangle [171]), à approcher la carte de densité d'un grand ensemble de points. Cette approche est l'homologue discret des cartes de densité lissées approximatives [133]. Comparé à ConcaveCubes [103] qui utilisent aussi des formes discrètes, nous nous sommes intéressées à l'établissement automatique d'ensemble de formes respectant un certain budget de stockage dans le but de pouvoir paramétrer le taux de compression de ces données géométriques.

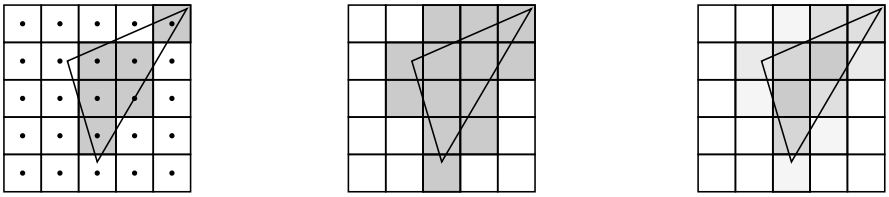
4.3 Préliminaires

Notre objectif est de constituer des simplifications géométriques des données permettant de reconstruire une carte de densité. Plus précisément, les images de rendu de ces simplifications doivent posséder des caractéristiques essentielles de la carte de densité de référence, c.-à-d. celle rendue à partir de tous les points. Pour un jeu de données et une définition donnée, il existe une unique image de référence et plusieurs autres images de différentes *qualités*, c.-à-d. différents degrés de ressemblance à l'image de référence, issues de sa compression à différents taux. De manière similaire et quelle que soit la définition, on peut constituer différentes « *approximations* » *géométriques* du jeu de données, de qualités différentes, qui pour une définition donnée, résultent en des images de qualités différentes qu'on qualifie ici d'images *abstraites*.

4.3.1 Approche conservatrice

Nous définissons la préservation de l'information par deux propriétés : la conservation de la *masse* et de la *couverture*. La préservation de la masse signifie que l'intensité totale des pixels est la même sur une image abstraite et sur son image de référence. La préservation de la couverture signifie que les pixels non vides sur la référence ne le sont pas sur l'image abstraite. Pour respecter ces deux propriétés, le processus de rendu doit être capable de contrôler l'intensité totale assignée aux pixels de l'image et de garantir que tous les pixels couverts par une forme géométrique soient colorés. Puisque la forme finale de présentation des données à l'utilisateur est l'image, la *qualité* d'une abstraction géométrique sera évaluée sous leur forme discrétisée par comparaison à l'image de référence.

Dans le pipeline classique de rendu graphique, l'étape de rastérisation transforme des *primitives* géométriques, définies par des sommets positionnés dans un espace continu en des *fragments*, eux-mêmes ensuite colorés pour former la séquence de pixels qui constitue l'image finale (cf. figure 4.6).



(a) Rasterisation par échantillonnage uniforme.

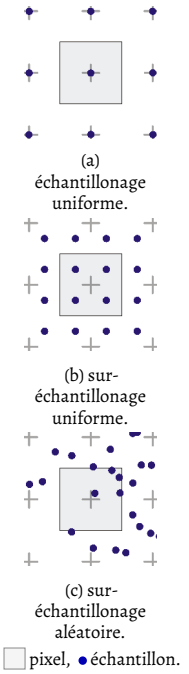
(b) Rastérisation conservatrice (surestimation).

(c) Calcul d'aire.

FIGURE 4.7 – Différents modes de rastérisation illustrés pour un triangle dont tous les sommets ont la même valeur d'attribut.

RASTÉRISATION PAR ÉCHANTILLONNAGE UNIFORME Sous sa forme la plus simple, la rastérisation utilise le centre de chaque pixel comme échantillon pour déterminer la couleur de chaque fragment (cf. figure 4.7a). Avec cette approche, les pixels partiellement couverts par la primitive sans que leur centre ne le soit ne sont pas colorés.

SUR-ÉCHANTILLONAGE Pour mieux représenter la primitive, le *sur-échantillonnage* consiste à utiliser plus d'un échantillon par pixel, la valeur du pixel étant ensuite calculée par agrégation des valeurs des échantillons du pixel. Deux stratégies de sur-échantillonnage sont illustrées sur la figure 4.8. En pratique, le sur-échantillonnage est utilisé pour réduire les artefacts de crénelage (*aliasing*), comme les lignes pixelisées, et ainsi sert à produire des images de synthèse plus agréables ou réalistes. Dans notre contexte par contre, nous recherchons une technique de rastérisation qui fournisse des garanties sur la somme totale de l'intensité des pixels et sur la conservation de la couverture entre l'image continue et sa forme discrète. L'échantillonnage et le sur-échantillonnage ne fournissent que des garanties probabilistes pour la couverture : les fragments qui ont au moins un échantillon couvert par la primitive sont colorés.



(a) échantillonage uniforme.

(b) sur-échantillonage uniforme.

(c) sur-échantillonage aléatoire.

pixel, échantillon.

FIGURE 4.8: Deux stratégies de sur-échantillonnage (b-c) par rapport à l'échantillonnage uniforme (a).

RASTÉRISATION *CONSERVATRICE* La rastérisation conservatrice est une approche non probabiliste qui colore tous les fragments au moins partiellement couverts par la primitive. et ainsi garantie que chaque pixel dont la région couvre la primitive sera coloré (cf. figure 4.7b). Cette approche préserve la couverture et est assurée matériellement [3].

CALCUL D'AIRE Pour éviter les effets de crénelage sur les bords, les méthodes par calcul d'aire consistent à colorer chaque pixel couvert par la primitive avec une intensité dépendant sur la surface de la zone couverte (cf. figure 4.7c). En général, le calcul de l'aire de cette zone par une méthode analytique a un coût prohibitif et il est donc approximé par subdivision des pixels en sous-pixels (sur-échantillonnage). Dans le cas de formes géométriques en 2D et de couleur uniforme, la méthode analytique est utilisable, bien qu'elle reste limitée en précision par celle des calculs flottants ou arithmétiques.

Afin de rendre des images répondant aux deux pré-requis de couverture et de préservation de la masse, nous avons implémenté un processus de rendu par calcul d'aire dont l'approche est décrite en section 4.4.3.

4.3.2 *Évaluation automatique par mesure de qualité*

Les mesures de qualité pour la visualisation servent en général à évaluer une vue pour un jeu de données. Ce sont des fonctions prenant en entrée une image ou une description d'une visualisation et en estiment la qualité relativement à un critère : quantité d'encombrement visuel (p. ex. [18]), ou capacité à permettre l'identification de motifs (p. ex. [115, 88]). BEHRISCH et al. [13] présentent un état de l'art de l'utilisation des mesures de qualité dans la visualisation en

3. OpenGL Extension #465, OpenGL ES Extension #228, https://www.khronos.org/registry/OpenGL/extensions/NV/NV_conservative_raster.txt

distinguant, d'une part les cas d'application dans le domaine des données et dans le domaine image et d'autre part, le niveau de granularité des mesures selon trois niveaux. Le niveau le plus bas s'intéresse à la perception de différences et le choix de variables visuelles. Le niveau intermédiaire concerne par exemple l'efficacité relative à une tâche ou la réduction d'encombrement visuel. Enfin le niveau le plus élevé mesure des quantités plus subjectives comme l'esthétique ou la fidélité d'une visualisation. Certaines mesures de qualité évaluent l'abstraction visuelle, c'est-à-dire la capacité de la représentation à représenter les données, notamment dans des contextes où l'ensemble des données ne peut pas être représenté (p. ex. [44, 36]).

Les mesures de qualité dans l'espace image (ou écran [88]) permettent de capturer la structure graphique de la visualisation. L'avantage de cette approche est que la mesure de qualité opère sur le même type de données que l'utilisateur, une image [13]. Dans l'espace image, la mesure de *similarité structurelle* [164] est utilisée pour évaluer l'identification des points isolés et des groupes dans les nuages de points [115], pour évaluer la qualité des cartes de densité approchées [133], et pour la comparaison d'algorithmes de compression d'images. Cette métrique est catégorisée comme une mesure *objective* de la qualité d'une image, c.-à-d. elle cherche à *prédire* la qualité telle que rapportée par des observateurs humains [163]. La mesure SSIM [4] (pour *Structural SIMilarity* [164]) a été originellement développée pour évaluer la similarité d'images déformées (p. ex. par compression) à leur image de référence. Elle a pour objectif de mesurer les différences perceptibles par un humain plutôt que les différences numériques pixel à pixel comme l'erreur quadratique moyenne (MSE). Ainsi, elle est conçue pour être relativement peu sensible aux changements de luminosité (intensité moyenne) et de contraste (variance de l'intensité), mais faire état des déformations de structure telles que le bruit et le flou.

L'index similarité SSIM est défini comme un produit pondéré de trois fonctions : une fonction de similarité de la luminosité (dépendante de l'intensité moyenne des images), une fonction de similarité du contraste (dépendante de l'écart-type des images), et une fonction de similarité structurelle. Avec les poids suggérés par les auteurs, son expression pour deux images X et Y, mono-canal et de même dimension est la suivante :

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}, \tag{4.1}$$

où μ désigne l'intensité moyenne, σ l'écart-type, σ_{XY} la covariance de X et Y et C_1 et C_2 sont deux constantes. La mesure de similarité utilisée pour comparer deux images est une moyenne glissante de cet index, calculée sur les N pixels :

$$MSSIM(X,Y) = \frac{1}{N}\sum_{j=1}^{N} SSIM(x_j, y_j), \tag{4.2}$$

où x_j et y_j désignent les patches formés par la fenêtre glissante centrée sur le j^e pixel [5]. La mesure *MSSIM* est symétrique et a ses valeurs dans $[0, 1]$ avec 1 désignant des images structurellement identiques. La dénomination *SSIM* fait en général référence à la mesure *MSSIM*, et cette convention sera adoptée dans ce chapitre et en particulier en section 4.6.1 qui l'utilise.

	SSIM
excel.	[0, 99; 1]
good	[0, 95; 0, 99[
fair	[0, 88; 0, 95[
poor	[0, 50; 0, 88[
bad	[0; 0, 5[

TABLE 4.2: Échelle nominale pour les scores SSIM [180]. La relation entre les scores SSIM et l'appréciation d'un humain n'est pas linéaire.

Les scores SSIM ont été expérimentalement comparés à des jugements humains sur un corpus de photographies compressées [164] ce qui a montré une forte corrélation, non linéaire, entre les scores SSIM et des notes d'opinion moyennes fournies par les participants de l'étude. Dans une autre étude, les scores SSIM ont été fait correspondre à des notes qualitatives [180]. Le tableau 4.2 rapporte la correspondance entre l'échelle des scores SSIM et cette échelle nominale mettant en évidence la relation non-linaire entre elles.

4. Structural SIMilarity

5. Dans l'implémentation des auteurs, les fenêtres ont pour dimensions 11×11 et sont pondérées par une gaussienne ($\sigma = 1, 5$)

4.4 Carte de densité simplifiée et conservatrice

Dans cette section nous présentons les notations utilisées, l'idée générale d'abstraction géométrique de taille bornée et l'approche de rastérisation utilisée. On s'intéresse à la représentation d'une séquence de n points, notée D, appartenant à un espace $\Omega \subset \mathbb{R}^2$. Les données géométriques correspondant à l'image de référence de D sont nommées *géométrie de référence* et notée G. La géométrie G de D est l'ensemble des points distincts de D, pondérés par leur occurrence :

$$G = \{(m, p) : p \in \Omega, m = \big|\{q \in D,\ q = p\}\big|\}. \tag{4.3}$$

4.4.1 Géométrique simplifiée conservatrice

Une *abstraction géométrique* de D, notée $\tilde{G}$, est un ensemble de formes géométriques pondérées qui vérifie les propriétés de préservation (1) de la masse et (2) de la couverture, relativement à G. Une *forme géométrique* est une région fermée arbitraire de l'espace Ω, qui peut être dégénérée en une ligne ou un point. $\tilde{G}$ est de la forme :

$$\tilde{G} = \{(m, s) : m \in \mathbb{R}_+^*,\ s \subset \Omega\}, \tag{4.4}$$

où s est une région, c.-à-d. une forme, et m un poids. Les deux propriétés de préservation mentionnées précédemment, s'expriment alors :

1. Préservation de la masse : $\sum_{(m,s)\in\tilde{G}} m = n$;
2. Préservation de la couverture : $\forall p \in D,\ \exists (m, s) \in \tilde{G},$ tel que $p \in s$.

En pratique, la somme des poids des formes couvrant une région arbitraire de Ω doit être lié au nombre de points de D couverts par ces formes. Puisque les formes géométriques peuvent se superposer, il n'est pas possible d'exprimer cette propriété par la préservation de la masse à l'échelle de chaque forme. Cette propriété est donc restreinte à une condition de majoration : le poids de chaque forme doit être majoré par le nombre de points inclus dans la région de la forme, c.-à-d. :

3. Masse des formes : $\forall (m, s) \in \tilde{G},\ m \leqslant \big|\{p \in s : p \in D\}\big|$.

Sous ces conditions, la géométrie de référence est un cas particulier d'abstraction géométrique où toutes les formes sont des points. Ces propriétés assurent que les abstractions d'un même jeu de données approchent la géométrie de référence à mesure que l'aire cumulée des formes qui la compose tend vers zéro.

La figure 4.9 montre un exemple de géométrie abstraite pour un ensemble de 13 points. L'abstraction géométrique présentée en bas à gauche est constituée de formes pondérées avec s_1 et s_3 couvrant chacune 6 points et s_2 n'en couvrant qu'un.

4.4.2 Taille d'une géométrie et taux de compression

La taille d'une abstraction correspond à son coût de stockage et dépend donc du type d'encodage utilisé. Dans cette section, nous résumons par simplicité cette taille au nombre de valeurs flottantes nécessaires pour définir les formes pondérées. Un point sera encodé par 2 valeurs, un cercle par 3, un rectangle dont les côtés sont parallèles aux axes par 4, un k-gone par $2k$, etc. Les formes peuvent être de types différents pour une même abstraction, tant que le client de visualisation est en mesure de reconstruire cette géométrie à partir des données fournies, c'est-à-dire de *décoder* la géométrie. Dans notre cas, le nombre de valeurs associées à chaque forme permet au client de déterminer son type de chaque forme (cf. tableau 4.3). On notera *taille* la fonction associant à une géométrie le nombre de valeurs nécessaires à son encodage selon ce schéma.

FORME	TAILLE
point	2
cercle	3
rectangle (aligné aux axes)	4
triangle	6
quadrilatère	8
...	...
k-gone	$2k$

TABLE 4.3: Règle de décodage pour chaque forme (non pondérée) dans le format compact utilisé pour notre mesure *taille* .

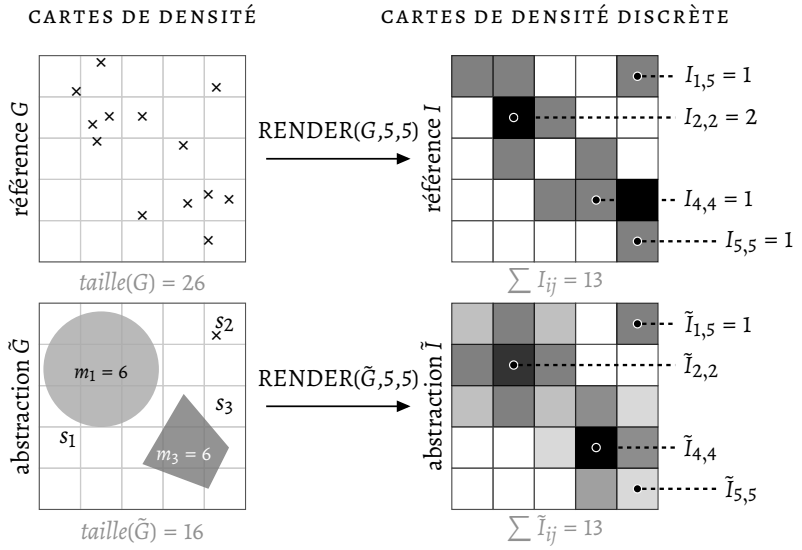


FIGURE 4.9 – Illustration de deux cartes de densité pour le même ensemble de 13 points, et de leurs cartes de densité discrète (CDD) pour une définition 5×5. L'intensité des pixels des CDD est signifiée par une échelle de niveaux de gris où les pixels foncés correspondent aux valeurs les plus élevées. Sur $\tilde{G}$, la différence d'intensité entre s_1 et s_3, qui ont le même poids, est la conséquence de leur différence d'aire. Sur $\tilde{I}$, la différence d'intensité entre les pixels $\tilde{I}_{4,4}$ et $\tilde{I}_{5,5}$, tous les deux couverts seulement par la forme s_3, est due au fait que $\tilde{I}_{4,4}$ a une intersection avec s_3 plus grande que celle de $\tilde{I}_{5,5}$.

Le taux de compression τ, en général exprimé en pourcentage, mesure le gain en espace procuré par une compression. Il est défini, par rapport aux données, comme suit :

$$\tau = 1 - \frac{taille(\tilde{G})}{taille(G)} = 1 - \frac{taille(\tilde{G})}{2n}, \tag{4.5}$$

en considérant que les poids unitaires de la géométrie d'entrée ne sont pas transférés. On exprime le taux de compression sous une autre forme comparant le transfert de la géométrie compressée à celui de son image. Cette expression du taux de compression, notée r, est définit comme suit :

$$r = \frac{taille(\tilde{G})}{w \cdot h}, \tag{4.6}$$

où $w \cdot h$ est la définition de l'image considérée. Ce ratio est une expression *théorique* du gain du transfert de la géométrie compressée puisqu'aucune compression d'image n'est considérée.

Sur l'exemple de la figure 4.9, l'abstraction géométrique du jeu de données de 13 points est composée d'un cercle s_1, d'un point s_2 et d'un quadrilatère s_3. Cette géométrie peut être encodée par 16 valeurs (13 pour les formes et 3 pour leurs poids) ce qui correspond à un taux de compression $\tau \simeq 38$ %. Cette estimation de la taille d'une géométrie en nombre de valeurs flottantes permet aussi d'estimer son espace de stockage nécessaire. Pour des valeurs flottantes codées sur 64 bits, on estimerait la taille réelle de stockage des 26 points à 208 octets et la taille réelle de l'abstraction à 128 octets, sans compression supplémentaire.

4.4.3 *Rendu préservant l'information*

Une *carte de densité* est un champ scalaire 2D, c.-à-d. une fonction associant une valeur scalaire à tout point de l'espace Ω. La carte de densité pour une abstraction géométrique correspond à son image dans l'espace continu. Afin de respecter la propriété de conservation de la masse, l'intégrale de cette carte de densité doit valoir la masse des données c.-à-d. la somme des poids

Entrée: w et h sont les dimensions de la CDD en sortie.
fonction RENDER($\tilde{G}$, w, h)
$\tilde{I} \leftarrow 0_{w,h}$
pour tout $(i, j) \in \{1 \dots w\} \times \{1 \dots h\}$ **faire**
pour tout $(m, s) \in \tilde{G}$ **faire**
$p \leftarrow regionPixel(i, j)$ ▷ région $p \subset \Omega$
$q \leftarrow 0$ ▷ facteur d'étalement du poids
▷ Déterminer le facteur d'étalement q en fonction de la forme
si *estUnPoint*(s) **alors** ▷ cas dégénéré n°1
$q \leftarrow \mathbb{1}_{s \in p}$ ▷ 0 si en dehors du pixel
sinon si *estUnSegment*(s) **alors** ▷ cas dégénéré n°2
$q \leftarrow \frac{aire(p \cap ligne(s))}{aire(ligne(s))}$
sinon
$q \leftarrow \frac{aire(p \cap s)}{aire(s)}$ ▷ 0 si $s \cap p = \emptyset$
▷ Mise à jour de l'intensité du pixel
$\tilde{I}_{ij} \leftarrow \tilde{I}_{ij} + mq$
return I

Algorithme 1 – Rendu d'une abstraction géométrique $\tilde{G}$ pour obtenir une CDD de taille $w \times h$.

des formes. Pour cela, chaque forme pondérée $(m, s) \in \tilde{G}$ est représentée dans la carte de densité par sa fonction indicatrice pondérée par le ratio entre son poids et son aire. Vu autrement, les formes étendent leur poids uniformément sur leur surface, comme illustré par la différence d'intensité des formes s_1 et s_3 qui ont le même poids sur la figure 4.9.

Effectuer le rendu d'une carte de densité implique de discrétiser à la fois ses valeurs spatiales et ses valeurs d'intensité. On appelle *carte de densité discrète* (CDD [6]) la matrice de valeurs réelles $I \in \mathbb{R}^{w \times h}$ résultant de la discrétisation spatiale d'une carte de densité à la définition $w \times h$. La discrétisation des valeurs d'intensité est en lien avec le choix des échelles de couleurs [16], sujet dont nous ne nous préoccupons pas dans ce travail puisque les valeurs d'intensité ne sont donc pas discrétisées dans une CDD.

Pour une définition $w \times h$, la CDD de référence d'un jeu de données D est obtenue par rastérisation de ses points. Ce processus est une agrégation, chaque pixel se voit affecter le nombre de points des données qui sont inclus dans la région unitaire de Ω qui lui correspond (cf. figure 4.9, en haut). Les propriétés de préservation de la masse et de la couverture définies pour une abstraction $\tilde{G}$ ont leur équivalent dans leur forme rendue $\tilde{I}$. Pour une définition $w \times h$, $\tilde{I}$ de référence I, ces propriétés sont :

1. Préservation de la masse : $\sum_{i \leqslant h} \sum_{j \leqslant w} I_{ij} = \sum_{i \leqslant h} \sum_{j \leqslant w} \tilde{I}_{ij}$
2. Préservation de la couverture : $\forall i \leqslant h,\ j \leqslant w,\ I_{ij} \neq 0 \Rightarrow \tilde{I}_{ij} \neq 0$

La préservation de la couverture implique qu'aucun pixel de $\tilde{I}$ n'est d'intensité nulle alors que son homologue dans I ne l'est pas. Sans ces faux-négatifs, la similarité de $\tilde{I}$ avec I augmente à mesure que son taux de faux-positifs diminue.

Pour rendre les cartes de densités en vérifiant ces propriétés, les abstractions géométriques sont discrétisées par calcul d'aire. Chaque forme géométrique contribue à l'intensité de chaque pixel dont elle chevauche la région, d'un facteur dépendant du poids de la forme et de l'aire du chevauchement. Cette méthode de rendu conservateur est décrite par l'algorithme 1 et illustrée sur la figure 4.9, par les différences d'intensité entre $\tilde{I}_{4,4}$ et $\tilde{I}_{5,5}$ d'une part et $\tilde{I}_{4,4}$ et $\tilde{I}_{2,2}$ d'autre part. Le premier cas est un exemple de deux pixels d'intensité différentes bien que recouverts par la même unique forme, s_3, puisque leur aire de chevauchement est différente. $\tilde{I}_{5,5}$ étant entièrement recouvert par s_3, il est d'intensité plus forte que $\tilde{I}_{4,4}$. Le second cas est un exemple de deux pixels d'intensités différentes bien que recouverts entièrement par des formes de même poids puisque les aires de leurs formes couvrantes sont différentes. La forme s_1 étant d'aire plus grande, $\tilde{I}_{2,2}$ a une intensité plus faible que $\tilde{I}_{4,4}$. La cas dégénéré des points est traité de la même manière que le rendu de la référence et celui des lignes est traité comme des quadrilatères. Cette

6. carte de densité discrète

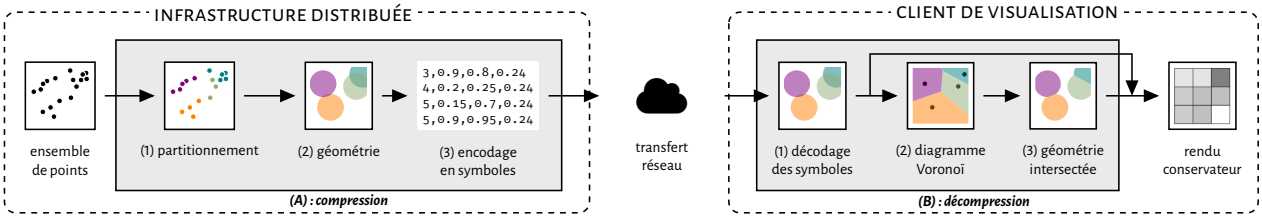


FIGURE 4.10: Vue d'ensemble du pipeline de compression géométrique, commun à toutes les stratégies.

approche cohérente retranscrit des portions d'images à l'identique là où la géométrie est aussi fine que les données. Par exemple, sur la figure 4.9, $\tilde{I}_{1,5} = I_{1,5}$.

4.5 Stratégies de compression géométrique

Comme expliqué précédemment, on s'intéresse à l'utilisation de réduction géométrique (ou *abstraction* géométrique) pour compresser des cartes de densités d'ensembles de points. Ces compressions par simplification géométrique sont des compressions *avec pertes*, respectant les propriétés conservatrices décrites dans la section précédente, et fonctionnant à partir d'un objectif de taux de compression, c.-à-d. sous une contrainte de budget pour la taille de la géométrie compressée. Cette approche permet de fournir des garanties sur les coûts de stockage et de transfert.

Il a de multiples manières de calculer des abstractions géométriques dans ce contexte. Par exemple, une stratégie cherchant à maximiser le nombre de formes peut utiliser exclusivement des cercles puisque ceux-ci ont l'empreinte de stockage la plus petite. Une stratégie cherchant à minimiser l'aire des formes peut utiliser des polygones englobants dont l'empreinte en stockage est plus forte mais qui entourent plus précisément les groupes de points que les cercles.

4.5.1 *Approche générale*

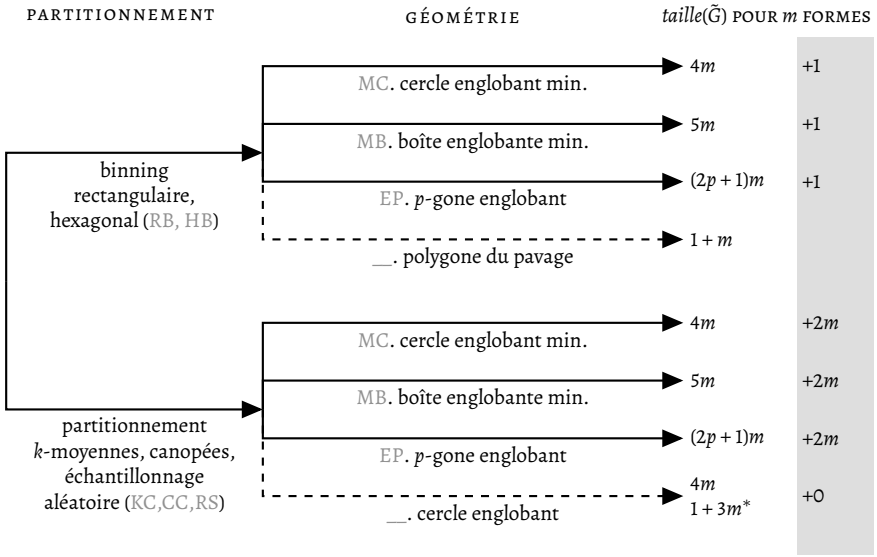
Nous nous intéressons à des stratégies formée de deux étapes : une première partitionnant les données, et une seconde dérivant des formes géométriques englobantes pour chaque partie. La figure 4.10 résume ce processus : l'étape de compression (A) inclut un partitionnement des points (A1), la constitution d'une géométrie pondérée à partie de la partition résultante (A2), et enfin l'encodage de la géométrie (A3) utilisant le format décrit dans la section précédente.

En combinant différentes méthodes de partitionnement et différentes représentations par formes géométriques élémentaires, on obtient de multiples stratégies présentées en figure 4.11. Les stratégies de compression décrites dans la suite de cette section sont des algorithmes prenant en entrée un taux de compression τ. Pour le respecter, la taille de la géométrie $\tilde{G}$ produite doit être majorée par le budget k équivalent τ, tel que :

$$taille(\tilde{G}) \leqslant k = 2n(1-\tau). \tag{4.7}$$

Sachant que les stratégies produisent des formes du même type, la taille de chacune est majorée par une valeur t et le budget k détermine ainsi le nombre m de parties fournit en paramètre à l'étape de partitionnement : $m = \lfloor \frac{k}{t} \rfloor$. Ainsi, on obtiendra :

$$taille(\tilde{G}) \leqslant mt \leqslant \left\lfloor \frac{k}{t} \right\rfloor \cdot t \leqslant k \tag{4.8}$$



- - - géométrie canonique
∗ pour le partitionnement par canopées : cercles de rayons identiques
coût additionnel de l'option de rognage Voronoï.

FIGURE 4.11 – Stratégies composées en associant une méthode de partitionnement à une méthode de géométrie avec ou sans le rognage Voronoï. Pour chaque stratégie, *taille*($\tilde{G}$) est exprimée en fonction de m, le nombre de parties formées à l'étape de partitionnement. Pour certaines stratégies, le rognage optionnel utilisant le diagramme de Voronoï induit un coût additionnel.

PARTITIONNEMENT		GÉOMETRIE		ROGNAGE VORONOÏ	
KC	*k*-moyennes	MC	cercle englobant minimal	VC	avec
CC	canopées	MB	boîte englobante minimale	__	sans
RS	échantillonnage aléatoire	EP	polygone englobant		
RB	*binning* rectangulaire	__	canonique		
HB	*binning* hexagonal				

TABLE 4.4 – Notations utilisées pour décrire les stratégies par leurs composantes. Ces notations suivent le format `[partition][géométrie][rognage]`, où chaque partie est un code à deux lettres.

On utilisera par la suite la notation compacte `[partitionnement] [géométrie] [rognage]` détaillée sur le tableau 4.4 pour dénommer ces stratégies composées par association de technique à différentes étapes.

4.5.2 *Partitionnement des points (A1)*

Pour la première étape de partitionnement des points, on considère cinq techniques usuelles produisant des ensembles disjoints de points. Chacune de ces méthodes est associée à une représentation géométrique canonique dont le coût est reporté sur la figure 4.11.

PARTITIONNEMENT *BINNING* Les méthodes par *binning* groupent tous les points appartenant à la même cellule d'un pavage régulier de l'espace composée de carrés ou hexagones réguliers. Leur géométrie canonique fait correspondre chaque partie à la région formée par sa cellule. Ces méthodes ont l'avantage d'être compactes à encoder : dans le cas d'un remplissage dense de la grille, seule la taille de la grille et les poids, possiblement nuls, des cellules sont nécessaires. Pour une grille de m cellules, la géométrie est alors de taille $1 + m$, où 1 correspond à la taille de la grille.

Partitionnement par k-moyennes et échantillonnage Le partitionnement en k-moyennes (k-*means*) fonctionne de manière itérative et assigne chaque point au centroïde le plus proche. Les m centroïdes sont initialement choisis aléatoirement parmi les points et ensuite déplacés au barycentre des points qui leur sont assignés à chaque itération. L'affectation lors de l'atteinte de l'équilibre détermine le partitionnement des points. Le partitionnement par échantillonnage sélectionne m représentants aléatoirement et sans considération spatiale, puis affecte à chaque point son représentant le plus proche pour former une partition des points. La représentation géométrique canonique pour ces deux partitionnements est un ensemble de m cercles englobants, centrés sur le centroïde ou représentant de chaque partie et dont le rayon est calculé comme la distance maximale entre ce point et les points de la partie. Cette géométrie a dont une taille de $4m$.

Partitionnement par canopées Le partitionnement par canopées est une version de l'algorithme de McCallum et al. [113] adaptée à l'agrégation géométrique par Perrot [131]. Les points sont parcourus un par un et sélectionné comme représentant s'ils sont plus loin d'une distance d de tous les représentants sélectionnés jusqu'alors. Une seconde passe sur les points leur assigne leur représentant le plus proche ce qui définit un partitionnement des points. La représentation canonique du partitionnement par canopées consiste en des cercles centrés sur les représentants de chaque partie et de rayon égal à la distance d'agrégation d utilisée. Cette géométrie a un coût de $1 + 3m$ valeurs pour m cercles.

4.5.3 Représentation par formes géométriques pondérée (A2)

En plus de ces représentations géométriques canoniques, on considère trois approches supplémentaires pour constituer des formes couvrantes à partir d'une partition des points et rapportons leur coût sur la figure 4.11.

Puisque les cercles ont l'empreinte de stockage la plus faible, la première de ces méthodes utilise les cercles englobants minimaux (MC) comme forme pour chaque partie. Pour m parties, la géométrie pondérée coûte $4m$ avec cette technique. La seconde de ces méthodes utilise les boîtes englobantes minimales (MB), c.-à-d. des rectangles alignés aux axes, dont l'empreinte en stockage est de 5 valeurs en incluant leur poids, soit $5m$ pour m formes. Enfin, la troisième méthode utilise des p-gones englobants (EP) construits par simplification de l'enveloppe convexe de chaque partie, jusqu'à ce que les polygones possèdent p côtés ou moins. Les p-gones englobants ont pour objectif de réduire l'aire des formes couvrantes en étant plus ajustés à l'enveloppe de chaque groupe de point que les cercles ou rectangles. Leur coût est de $2p + 1$ valeurs par polygone pondéré, ce qui en fait, très largement, la méthode la plus coûteuse ($9m$ pour des quadrilatères quelconques soit plus de deux fois plus coûteux que les boîtes englobantes).

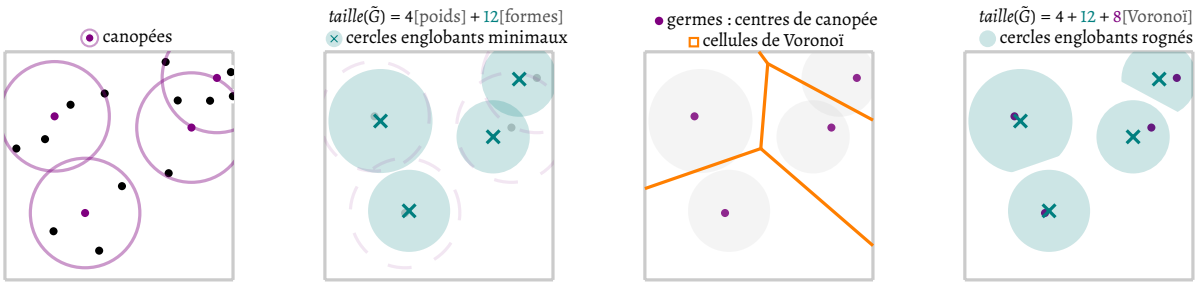
	MC	MB	EP	__
RB	■	■	■	
HB	■	■	■	
KC	■	■	■	■
CC	■	■	■	■
RS				

Table 4.5: Disponibilité de l'option rognage Voronoï en fonction du partitionnement (ligne) et de la représentation géométrique (colonne).

4.5.4 Reconstruction avec rognage par diagramme de Voronoï (B2-3)

Les cinq méthodes de partitionnement produisent des parties qui possèdent un point centroïde tel que chaque point du groupe est plus proche de ce centroïde que de tous les autres points. Ce centroïde n'est pas nécessairement un point des données ni l'unique point possédant cette propriété. C'est le centre des cellules issues du *binning*, les centroïdes des k-moyennes, les centres de canopées et les points élus pour l'échantillonnage aléatoire. De part leur propriété, ces points peuvent servir à calculer un diagramme de Voronoï divisant le plan et en conséquence les points des données, en accord avec la partition établie lors de l'étape de partitionnement (cf. figure 4.12c). Ce partitionnement spatial peut être utilisé pour raffiner les formes géométriques reconstruites du côté client, en les rognant pour ne conserver que l'intersection entre chaque forme et sa cellule du diagramme correspondante.

Le rognage n'est pas utilisé car inutile ou non défini dans deux cas (cf. tableau 4.5) :



(a) Partitionnement par canopées. (b) Sans rognage. (c) Diagramme de Voronoï. (d) Avec rognage.

FIGURE 4.12: Illustration de l'option de rognage par diagramme Voronoï pour le partitionnement par canopée et la représentation par cercles englobant minimal. Le partitionnement par canopées (a) définit quatre sous-ensembles des points, chacun représenté par un cercle englobant minimal sur (b). Cette géométrie sans rognage coûte quatre valeurs par forme, dont une pour le poids. Pour rogner les cercles, on calcule le diagramme de Voronoï en utilisant comme germes les centres des canopées (c). La représentation avec rognage (d) est composée des formes résultant de l'intersection de chaque cercle avec sa cellule de Voronoï correspondante. Cette représentation coûte deux valeurs de plus par forme, pour transmettre les centres des canopées qui ne sont pas les centres des cercles englobants.

— Partitionnement par *binning* avec représentation canonique : le diagramme de Voronoï utilisant les centres des cellules est identique aux formes résultantes du partitionnement.
— Échantillonnage aléatoire : le diagramme de Voronoï est non défini s'il existe des chevauchements dans les points de l'échantillon.

Le rognage Voronoï sert à réduire l'aire cumulée des formes tout en éliminant les chevauchements entre formes ce qui améliore selon notre hypothèse la qualité de la carte de densité. Dans le cas général, il coûte au plus un transfert de deux valeurs supplémentaires par forme (cf. figure 4.11). Pour les partitionnements par *binning*, seule l'échelle de la grille est nécessaire, soit une valeur. Pour les k-moyennes et canopées utilisant leur représentation canonique, les germes du diagramme de Voronoï sont les centres des cercles. Par conséquent, appliquer le rognage est gratuit. On peut donc supposer que dans ce cas, à budget égal l'option avec rognage soit supérieure.

On peut émettre une hypothèse similaire concernant les représentations canonique et par cercles englobants minimaux pour les partitionnements k-moyennes, canopées et échantillonnage aléatoire : la représentation canonique et celle par cercles englobants minimaux ont le même coût pour ces partitionnements. Puisque la représentation par cercles englobants minimaux réduit l'aire cumulée des formes, on peut supposer qu'à budget égal elle est supérieure.

De part leur partitionnement et leur méthode de dérivation de géométrie, les stratégies proposées font différents compromis pour approximer la distribution des points. Certaines stratégies tendent à produire de nombreuses formes géométriques (p. ex. partitionnement par *binning*) alors que d'autres tendent à en produire moins mais d'aire plus faible car plus ajustées au contour des parties de points (p. ex. polygones englobants). Dans la section suivante, on cherche à déterminer lesquelles de ces stratégies sont les plus *efficaces* en moyenne.

4.6 Évaluations des stratégies de compression géométrique

L'*efficacité* d'une stratégie est définie comme la capacité à produire des géométries dont la carte de densité discrète (CDD) résultante est la plus fidèle possible, c.-à-d. la plus similaire à sa référence, compte tenu d'une contrainte de taux de compression. La comparaison des stratégies repose sur l'évaluation de la fidélité de compression géométrique pour différents jeux de données et à de multiples taux de compression. Les taux de compression correspondent aux différents niveaux de la pyramide dans le cas de la navigation multi-échelle (cf. section 4.1).

Nous conduisons deux études pour comparer l'efficacité des stratégies décrites dans la section précédente. La première étude (section 4.6.1) est *objective*, elle utilise une mesure de similarité

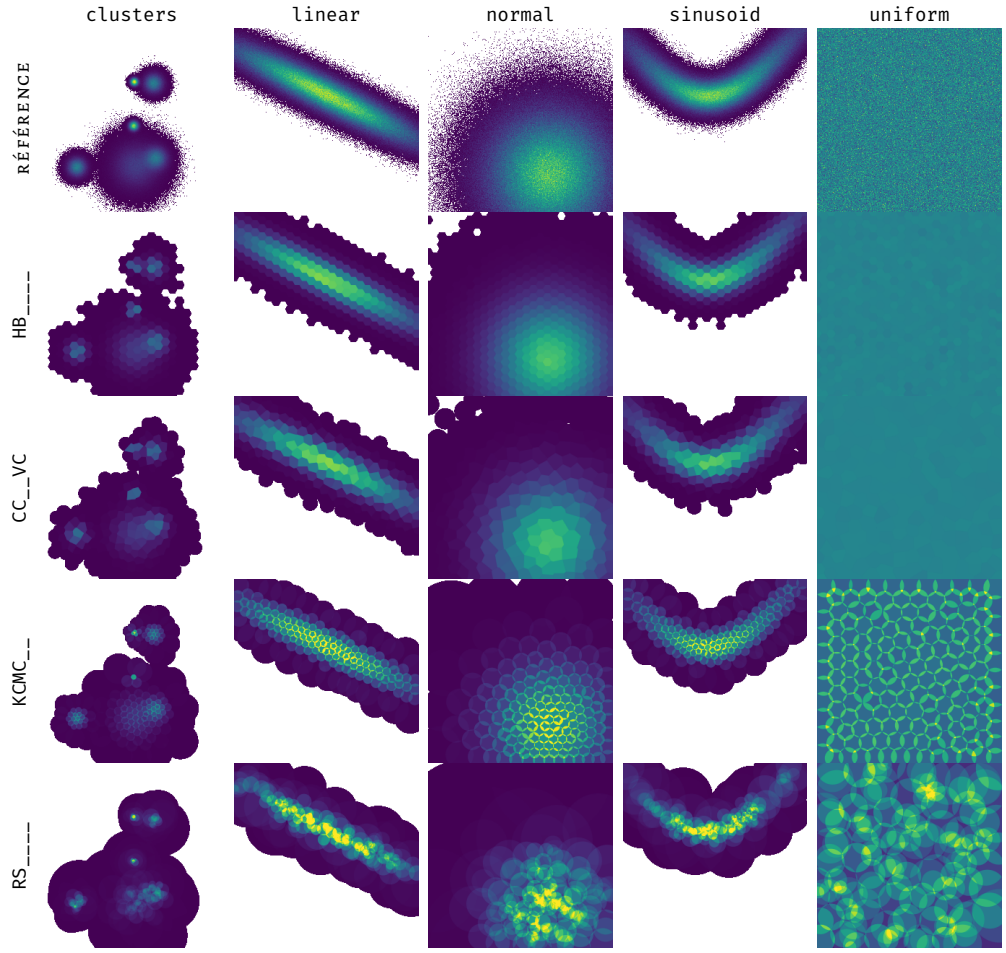


FIGURE 4.13 – Exemples de CDD (256 × 256), compression à $r = 1\%$, pour quatre stratégies et une instance de chaque distribution. Compte tenu de la taille des jeux de données générés relativement à la définition des images, les CDD pour la distribution `uniform` tendent à ne comporter aucun pixel nul.

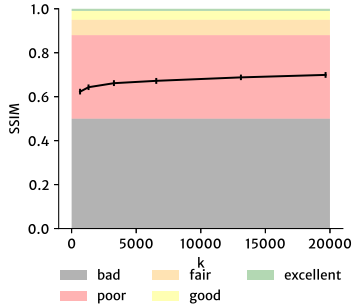
pour évaluer la qualité des CDD en comparaison à leur référence. Elle est précédée d'une étude préliminaire étudiant l'ensemble des stratégies. La seconde étude (section 4.6.2) est *subjective*, elle s'appuie sur des jugements humains pour comparer la fidélité d'images de CDD.

Les deux études utilisent des jeux de données à $n \in [10^6, 10^8]$ points et des CDD de définition 256 × 256 pixels de sorte que la densité moyenne de points par pixel soit supérieure à 10 ($n \gg w \times h$). On s'intéresse à l'efficacité des stratégies à produire des géométries de qualité suffisante à des taux de compression élevés relativement à la taille de l'image soit $r < 100\%$.

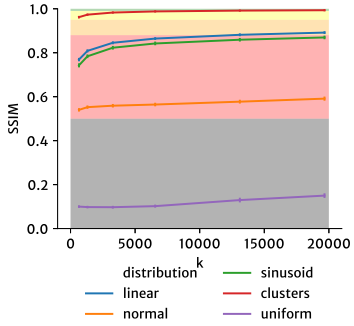
4.6.1 Comparaison des stratégies par métrique

Nous considérons les 34 stratégies résultant de la combinaison des partitionnements et représentations présentés dans la section précédente (cf. figure 4.11) :

— cinq méthodes de partitionnement : `RB`, `HB`, `KC`, `CC`, et `RS` ;
— quatre méthodes de dérivation de géométrie pour chaque partitionnement : `MC`, `MB`, `EP` (réduit au cas des quadrilatères, c.-à-d. $p = 4$), et `__` pour les méthodes canoniques ;
— une option de rognage Voronoï pour chaque couple à l'exception de ceux issus de `RS` et ceux issus du *binning* canonique (`RB__`, `HB__`).



(a) Scores SSIM pour l'ensemble des jeux de données.



(b) Scores SSIM pour chaque distribution de jeu de données.

distribution	Spearman
uniform	0,26
clusters	0,77
sinusoid	0,6
normal	0,23
linear	0,73
overall	0,16

(c) Coefficient de corrélation ($p \ll 0,01$)

FIGURE 4.14: Score SSIM moyennés sur toutes les CDD en fonction du budget k. (a) Relation entre le score SSIM moyen et le budget alloué k pour tous les jeux de données et toutes les stratégies. (b) Relation entre le score SSIM moyen et le budget alloué k pour chaque distribution. L'échelle nominale d'opinion humain pour le score SSIM est indiqué par les strates colorées. Les barres d'erreur correspondent à un intervalle de confiance à 95% pour la moyenne. (c) Coefficients de corrélation de Spearman entre les scores SSIM et le budget alloué, pour l'ensemble (*overall*) et pour chaque distribution.

DONNÉES D'ÉVALUATION ET CONDITIONS Nous choisissons six valeurs de budget k correspondant à des tailles de géométrie basses relativement à la taille de l'image ($r < 50\%$). Le tableau 4.6 liste les équivalences entre ces budgets et les taux de compression relativement à la taille des données (τ) et des images (r). Des jeux de données synthétiques d'un million de points sont générés en 30 instances pour chacune des cinq distributions proposées par SARIKAYA et al. [142] : `uniform`, `linear`, `clusters`, `sinusoid`, et `normal` (cf. exemples en figure 4.13). Cela revient à 150 jeux de données pour lesquels une CDD est générée pour chaque budget et chaque stratégie comme résumé sur le tableau 4.7. L'évaluation des stratégies par métriques se décompose en trois parties : la première est une étude préliminaire sur l'ensemble des données (E1), la seconde est une comparaison systématique des stratégies à budget égal (E2) et la dernière est une étude sur les stratégies de *binning* standard (E3).

k	τ	r
655	99,97	1
1310	99,94	2
3276	99,84	5
6553	99,67	10
13107	99,35	20
19660	99,02	30

TABLE 4.6: Équivalences entre le budget k, le taux de compression de la géométrie τ (%), et r (%) pour $n = 10^6$ et $w = h = 256$.

5	distributions
× 30	instances
150	jeux de données
× 6	budgets
× 34	stratégies
30600	CDD

TABLE 4.7: Conditions de l'étude comparative par métrique.

Étude préliminaire (E1)

Avant de comparer toutes les stratégies, nous avons évalué l'efficacité de l'approche générale proposée en s'intéressant à la relation entre le budget k et la qualité des CDD, toutes stratégies confondues.

PROTOCOLE La qualité de chaque CDD est évaluée par mesure de sa similarité à sa CDD de référence (une par jeu de données) en utilisant le score SSIM présenté en section 4.3.2. Le traitement des canaux de couleur par la métrique n'importe pas puisque les CDD ne possèdent pas de couleurs mais une seule valeur réelle d'intensité par pixel.

RÉSULTATS La figure 4.14 montre le score SSIM moyen en fonction du budget k, pour l'ensemble et pour chaque type de distribution, ainsi que les coefficients de corrélation entre les scores SSIM et le budget. La qualité des CDD tous budgets et distributions confondus, est faible en moyenne et la corrélation entre la qualité et le budget k est très faible dans l'ensemble. On fait les mêmes observations pour les CDD issues des distributions `normal` et `uniform` avec la qualité moyenne des CDD de `uniform` même pire que la moyenne ($SSIM < 0,3$). Pour les distributions `sinusoid`, `linear` et `clusters` par contre, la qualité est meilleure en moyenne ($SSIM > 0,7$) et la corrélation entre qualité et budget est forte. D'autre part, l'allure des courbes pour ces distributions indique que la qualité augmente rapidement initialement (budgets les plus bas), et augmente bien moins rapidement ensuite. Le point charnière se situe avant $k = 3000$

(r = 5%). Les meilleurs résultats sont obtenus par la distribution `clusters` pour laquelle les CDD sont de qualité bonne à excellente en moyenne pour tous les budgets.

Discussion De manière générale, cette étude préliminaire confirme nos attentes : les abstractions géométriques de taille bornée fournissent un mécanisme pour générer des CDD approchant progressivement la référence à mesure que le budget augmente. Elle montre également que les stratégies testées atteignent, en moyenne, une qualité acceptable (*fair*) dès r = 20% pour `linear` et une bonne qualité (*good*) dès le premier budget r = 1% pour `clusters`. La corrélation positive entre la qualité des CDD et leur budget est cependant plus faible qu'attendue. Cela peut être dû au fait que le budget est en relation avec la taille de la géométrie sans être nécessairement corrélé à leur aire cumulée. En effet, bien que toutes les stratégies produisent un nombre croissant de formes à mesure que le budget alloué grandit, le taux auquel l'aire des formes décroît peut varier entre stratégies.

Les différences de résultats entre distributions sont probablement dues à leurs caractéristiques propres. Par exemple la quantité moyenne de pixels d'intensité nulle sur les références est très faible voire nulle pour les jeux de données `uniform` et plus importante pour `clusters`, `linear` et `sinusoid` (cf. figure 4.13). En effet, les larges zones de pixels d'intensité nulle ont une forte probabilité d'être exactement retranscrites par la plupart des stratégies, même à faible budget. C'est particulièrement vrai pour les stratégies fondées sur un partitionnement par canopées ou *binning* dont l'étendue des formes est bornée par une certaine taille, décroissante avec l'accroissement du budget. En revanche, les stratégies fondées sur un partitionnement par k-moyennes ou échantillonnage aléatoire tendent à produire des formes plus étendues dans les zones de faible densité comme l'illustre l'apparence des zones extérieures pour les stratégies `KCMC__` et `RS____` sur la figure 4.13.

Comparaison des stratégies deux à deux (E2)

Les mêmes données d'évaluation ont ensuite été utilisées pour conduire une comparaison systématique des stratégies deux à deux.

Hypothèses Le principe général de compression géométrique conservatrice fait l'hypothèse que la couverture et la masse doivent être préservées et que la qualité d'une compression géométrique augmente à mesure que l'aire cumulée de ses formes approche zéro. On s'attend donc à ce que, à budget égal, les stratégies résultant en une aire cumulée plus faible résultent en des cartes de densités plus fidèles. Cette hypothèse concerne deux cas, évoqués dans la section précédente. Le premier est celui du rognage Voronoï, sans sur-coût pour `CC__` et `KC__` et à faible coût pour les *binning* avec géométrie non-canonique (cf. figure 4.11). Le second est celui des cercles englobants étant sans sur-coût relativement à la géométrie canonique pour les stratégies basées sur `RS` et `KC` (cf. figure 4.11).

Notre seconde hypothèse concerne la capacité des stratégies à retranscrire des détails de la densité des données. Les stratégies issues du partitionnement par canopées ou *binning* utilisant leur représentation canonique résultent en des formes géométriques de taille homogène et conditionnée par le budget. Au contraire, les stratégies utilisant le partitionnement par k-moyennes et l'échantillonnage aléatoire sont plus *adaptatives* sur cet aspect et susceptibles, à budget égal, de décrire plus fidèlement les détails les plus fins lorsque le budget est bas. Cette hypothèse est à mettre en relation avec le théorème d'échantillonnage qui énonce que « la représentation discrète d'un signal exige des échantillons régulièrement espacés à une fréquence d'échantillonnage supérieure au double de la fréquence maximale présente dans ce signal ». Le cas du partitionnement par canopées diffère cependant des partitionnements par *binning* car les centres de ses « cellules » ne sont pas nécessairement répartis de manière régulière dans l'espace. Ainsi, le client est capable de rendre un point à la place d'un cercle pour toutes les cellules de poids unitaire. De plus, l'aire cumulée des cercles peut être réduite par l'utilisation du rognage, à coût constant pour `CC__VC` notamment. On suppose donc que `CC__VC` présente un bon compromis entre faible coût par forme et régularité du partitionnement. Notre dernière hypothèse est donc que `CC__VC` résulte en

des résultats équivalents ou meilleurs que les stratégies de *binning* standard HB____ et RB____, en particulier pour les jeux de données présentant des points isolés.

Nous avons donc les hypothèses suivantes :

H1 À budget égal, les stratégies résultant en une plus faible aire cumulée des formes géométriques à coût constant surpassent les autres (cas du rognage Voronoï et des cercles englobants) ;

H2 À budget faible, les stratégies issues des partitionnements KC ou RS surpassent les autres ;

H3 CC__VC surpasse les stratégies de *binning* standard.

Protocole L'efficacité des stratégies est comparée paire à paire via SSIM, pour tous les jeux de données et tous les taux de compression, puis séparément pour les six taux de compression. Pour comparer les performances d'une paire de stratégies (A, B), nous calculons le nombre de couples *(jeux de données, budget)* pour lesquels les scores de qualité pour A sont supérieurs à ceux pour B.

Résultats Les scores de qualité SSIM ne sont pas distribués normalement selon un test de Shapiro-Wilk, pour l'ensemble des CDD (niveau de significativité $\alpha = 0,05$) ni pour les CDD séparées par budget (correction de Bonferroni : $\alpha' = 0,05/6 \simeq 0,008$). Les différences de performance entre stratégies sont mises en évidence par un test de Friedman à la fois pour tous les budgets ($\alpha = 0,05$, cf. figure 4.15) et par budget ($\alpha' = 0,05/6 \simeq 0,008$, cf. tableau A.1).

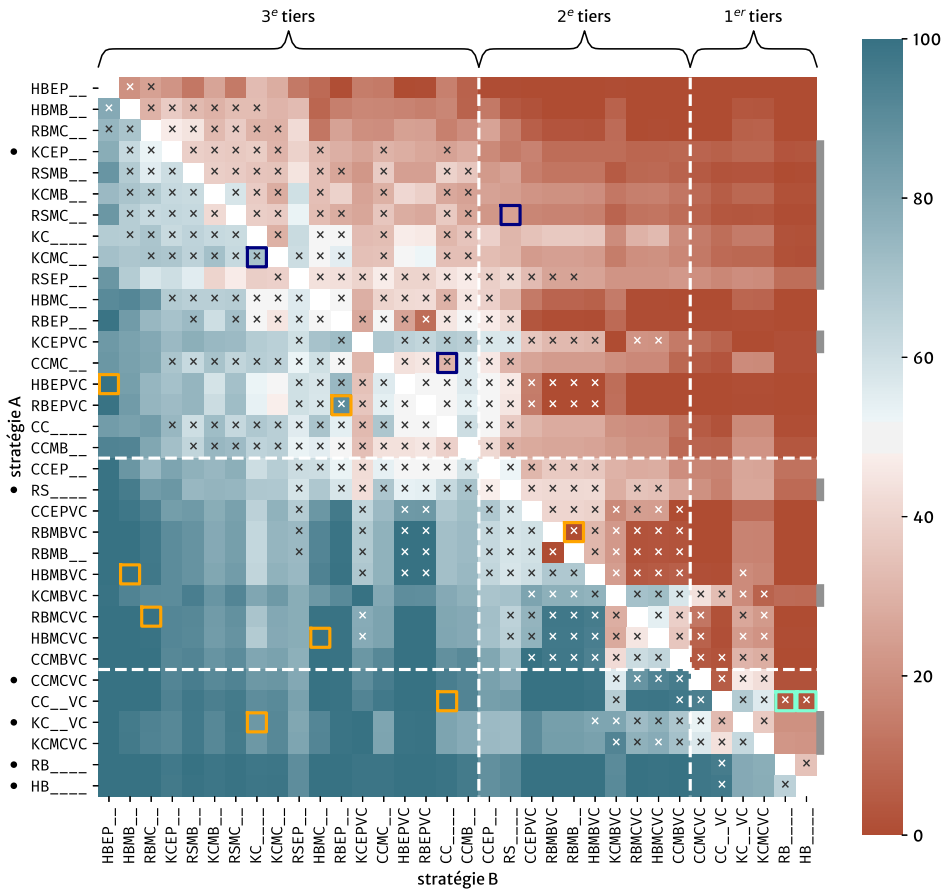
Les résultats des comparaisons paire à paire des stratégies sont rapportés sur la figure 4.15 pour l'ensemble des données et sur la figure 4.16 pour les six budgets séparément. La cellule (A, B) de chaque matrice rapporte la proportion de cas tels que A est plus performante que B (échelle de couleurs) ainsi que la significativité de la différence (annotation $\times$). Ainsi, les cellules bleues indiquent qu'A présente des résultats supérieurs à B pour plus de 50% des cas. Les cellules biffées indiquent que la différence de résultats entre A et B n'est pas significative (test de Conover). Sur l'ensemble des matrices, les stratégies sont ordonnées selon leur score SSIM moyen sur l'ensemble des données d'évaluation. Le classement est croissant de HB____ (bas) à HBEP__ (haut).

La significativité des résultats est cohérente, en général, avec les comparaisons : la majorité des cellules biffées correspondent à des valeurs autour de 50%. La plupart des stratégies utilisant le rognage Voronoï (VC) sont placées sur la partie inférieure (meilleure) de la figure 4.15, à l'exception des stratégies EPVC qui sont dans la partie supérieure. Les matrices par budget montrent que le classement général est plus cohérent avec celui des budgets les plus élevés (5%, 10%, 20%).

Discussion Afin de faciliter les comparaisons et discussions, nous avons séparés subjectivement les stratégies en trois groupes suggérés par l'allure de la matrice de la figure 4.15 et reporté ces catégories sur toutes les matrices. Le 1er tiers correspond aux stratégies les plus performantes et est particulièrement bien défini sur la matrice du budget le plus élevé ($r = 30\%$).

H1 Le premier cas de l'hypothèse H_1 est relatif au rognage Voronoï (VC) sans sur-coût et à bas sur-coût. L'hypothèse est vérifiée pour les stratégies CC__VC et KC__VC pour tous les budgets excepté le premier ($r = 1\%$) et dans la grande majorité des cas pour les stratégies par *binning* avec une géométrie non-canonique (cf. cellules □). Le second cas de l'hypothèse est relatif à l'utilisation de cercles englobants minimaux (MC) sans sur-coût. Il n'est pas vérifié dans l'ensemble ni par budget (cf. cellules □). Il est possible que le gain en aire du cercle englobant minimal par rapport à un cercle englobant soit bien trop faible pour avoir une incidence quelconque. L'hypothèse n'est pas non plus vérifiée pour les canopées pour lesquels les cercles englobants minimaux correspondent à un faible sur-coût.

H2 L'hypothèse H_2 concerne l'efficacité des stratégies RS et KC à bas coût. Les rangs concernés sont annotés sur les matrices (▮). Les résultats montrent deux cas différents. D'une part KC____ ou RS___ présentent de meilleures performances à bas budget qu'à haut budget. D'autres part, d'autres stratégies comme KCMBVC ou KCEPVC montrent l'inverse. Cela suggère que l'avantage potentiel des k-moyennes et de l'échantillonnage à bas budget est eclipsé par le coût des géométries non-canoniques.



Friedman : $\chi^2(34) = 17856,70$, $p \ll 0,05$
Annotations relatives aux hypothèses : ▫, ▫ H_1 • ▪ H_2 • ▫ H_3

FIGURE 4.15 – Matrices de comparaison des 34 stratégies à partir des scores SSIM. Les lignes et colonnes des matrices ont le même ordre et la couleur des cellules dépend du pourcentage de cas pour lesquels la stratégie en ligne surpasse la stratégie en colonne. Les cellules biffées (×) indiquent les différences non significatives ($\alpha' = \frac{0,05}{561} \simeq 0,00009$). Les stratégies pointées (●) sont celles utilisées dans l'étude utilisateur.

H_3 RB____ et HB____ surpasse en général toutes les stratégies, à l'exception de CC__VC (cf. cellules ▫). L'efficacité relative de ces stratégies peut provenir de la faible empreinte de stockage de leurs formes qui autorise un plus grand nombre de formes à budget égal. Cet effet peut permettre d'atteindre une granularité d'agrégation plus fine, lorsque les zones de vides ne sont pas trop importante pour que cet effet soit contrebalancé. L'hypothèse H_3 n'est pas vérifiée puisque CC__VC ne surpasse ni HB____ ni RB____. Cependant, les différences n'étant pas significatives, CC__VC peut être considérée comme une alternative compétitive.

Les stratégies KC__VC, KCMCVC, CCMCVC présentent également de bons résultats et bien que significativement moins performantes que les stratégies de *binning* en général, elles ne le sont pas aux trois budgets les plus élevés (cf. figure 4.16). De manière générale, les stratégies utilisant le rognage se démarquent aussi, notamment pour le budget le plus élevé ($k = 1960$, $r = 30\%$) comme l'indique la différentiation progressive d'une large zone bleue à partir de la troisième valeur de budget ($k = 3276$, $r = 5\%$) sur la partie inférieure où sont listées ces stratégies. En particulier, les stratégies utilisant les k-moyennes ont de mauvaises performances exceptées leur version

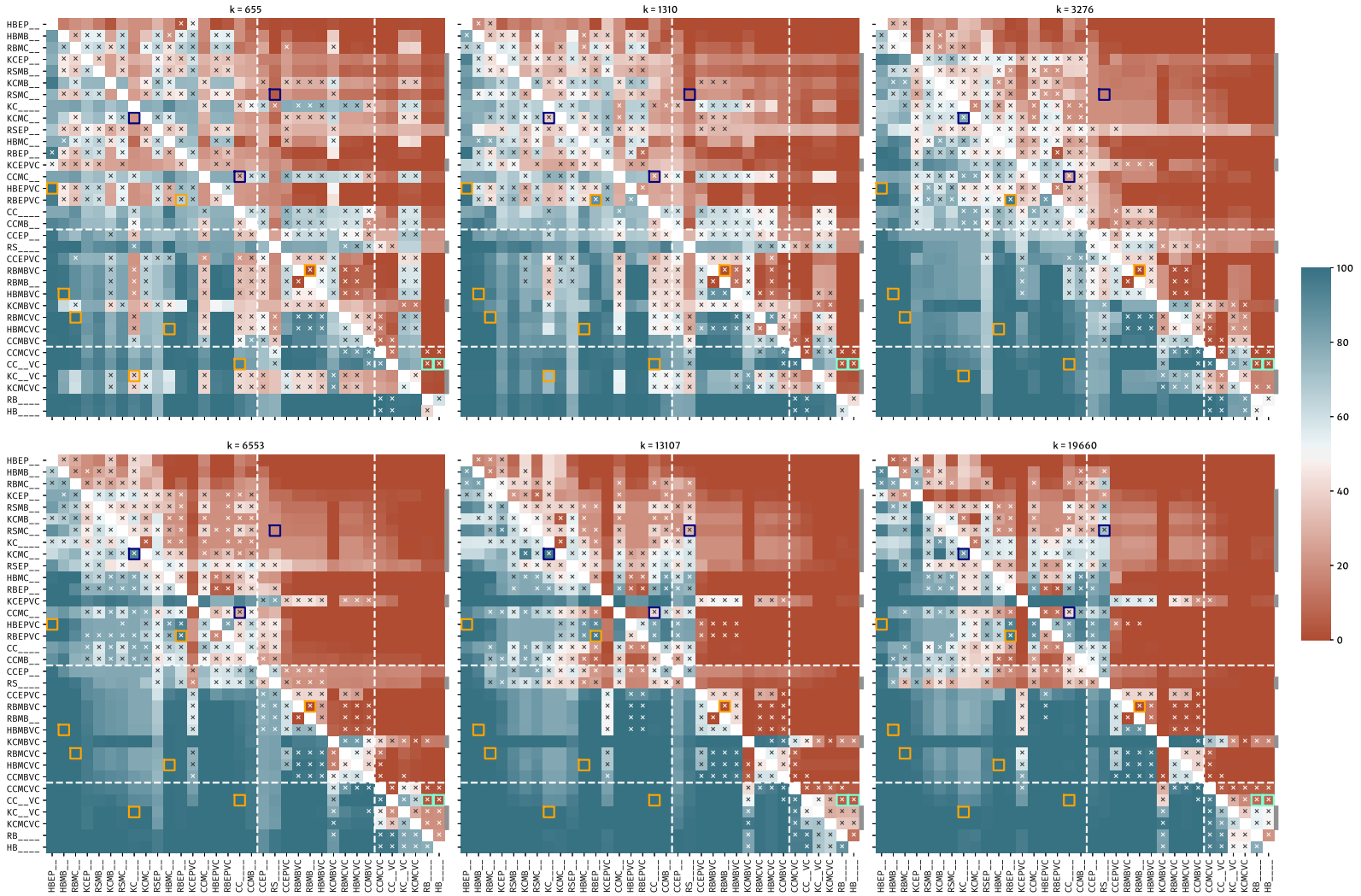
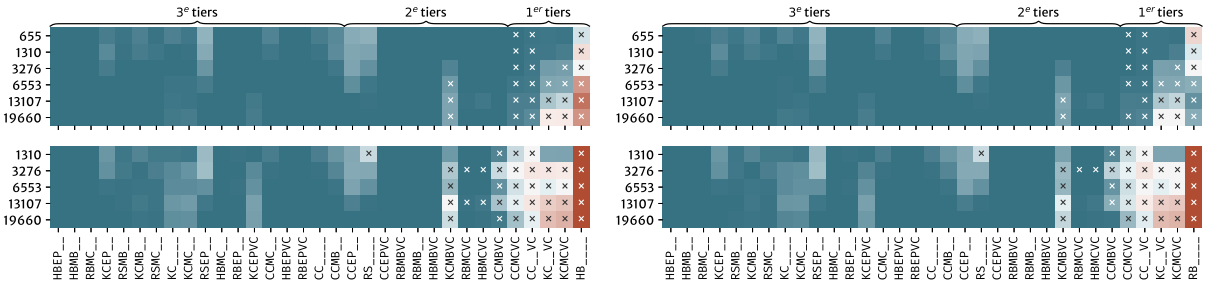


FIGURE 4.16 – Matrices de comparaison des 34 stratégies à partir des scores SSIM pour les différents budget k (croissant du coin supérieur gauche au coin inférieur droit). Les lignes et colonnes des matrices ont le même ordre et la couleur des cellules dépend du pourcentage de cas pour lesquels la stratégie en ligne surpasse la stratégie en colonne. Les cellules biffées (×) indiquent les différences non significatives ($\alpha' = \frac{0,05}{6 \cdot 561} \simeq 0,000015$). Annotations relatives aux hypothèses : □, □ H_1 • ▮ H_2 • □ H_3.



(a) RB____. (b) HB____.

FIGURE 4.17: Matrices de comparaison à budget inégal pour les deux stratégies par *binning*. La stratégie $S \in \{$RB____, HB____$\}$ à toutes les autres, en lignes. Les matrices supérieures rapportent les comparaisons à budget égal pour tous les budgets. Les matrices inférieures rapportent les comparaisons entre les stratégies des colonnes pour le budget indiqué en ligne et S à la valeur de budget précédente. Ainsi, la première ligne compare S à $k = 655$ aux autres stratégies à $k = 1310$. Ces dernières comparaisons sont donc en défaveur de S et serve à estimer l'écart d'efficacité. L'échelle utilisée est la même que sur la figure 4.15 : une cellule bleue indique que S surpasse la stratégie de la colonne et une cellule rouge l'inverse.

utilisant le rognage comme KCMBVC, KCMCCV, et KC__VC qui se distinguent et s'améliorent aux plus hauts budgets. L'élimination du chevauchement opérée par le rognage Voronoï pourrait être un facteur déterminant et complémentaire à la réduction de l'aire cumulée des formes.

Comparaison à budget inégal pour les stratégies de binning *(E3)*

Pour évaluer la différence d'efficacité entre les stratégies de *binning* et les autres, nous avons conduit dans un second temps des comparaisons par paire à budget *inégal*. L'objectif est de comparer l'écart d'efficacité en déterminant si, lorsque qu'une stratégie A surpasse une stratégie B au budget k_i, cette tendance persiste lorsque A au budget k_{i-1} est comparée à B au budget k_i (avec $k_{i-1} < k_i$).

La figure 4.17 présente les résultats de comparaison inégale pour RB____ et HB____ sur les matrices inférieures, avec les matrices supérieures rapportant les résultats de comparaison à budget égal (identiques aux résultats des matrices de la figure 4.15). Les matrices inférieures rapportent les valeurs de comparaison entre les stratégies des colonnes et la stratégie de comparaison $S \in \{$ RB____, HB____ $\}$.

DISCUSSION La comparaison à budget inégal fait ressortir la plupart des stratégies du 2e et 3e tiers comme étant fortement sous-efficaces relativement à RB____ et HB____ même dans cette comparaison inégale et défavorable pour RB____ et HB____. Les exceptions du 2e tiers sont KCMBVC et CCMBVC. Ces résultats suggèrent donc une forte différence de performances entre les stratégies du 1er tiers et les autres.

Les résultats de ces trois études montrent tout d'abord que, toutes stratégies confondues, l'augmentation du budget résulte en une augmentation du score de similarité moyen à la référence. Cette corrélation est la plus marquée pour les distributions `linear`, `sinusoid` et `clusters`. Pour ces distributions, les scores de similarité sont élevés même aux taux de compression les plus élevés. La comparaison systématique des stratégies a permis d'en identifier six se démarquant des 34 comparées (cf. figure 4.15). Parmi elles, on retrouve les *binning* standard (RB____ et HB____) et d'autres stratégies, issues des partitionnements par canopées ou k-moyennes et utilisent le rognage par Voronoï.

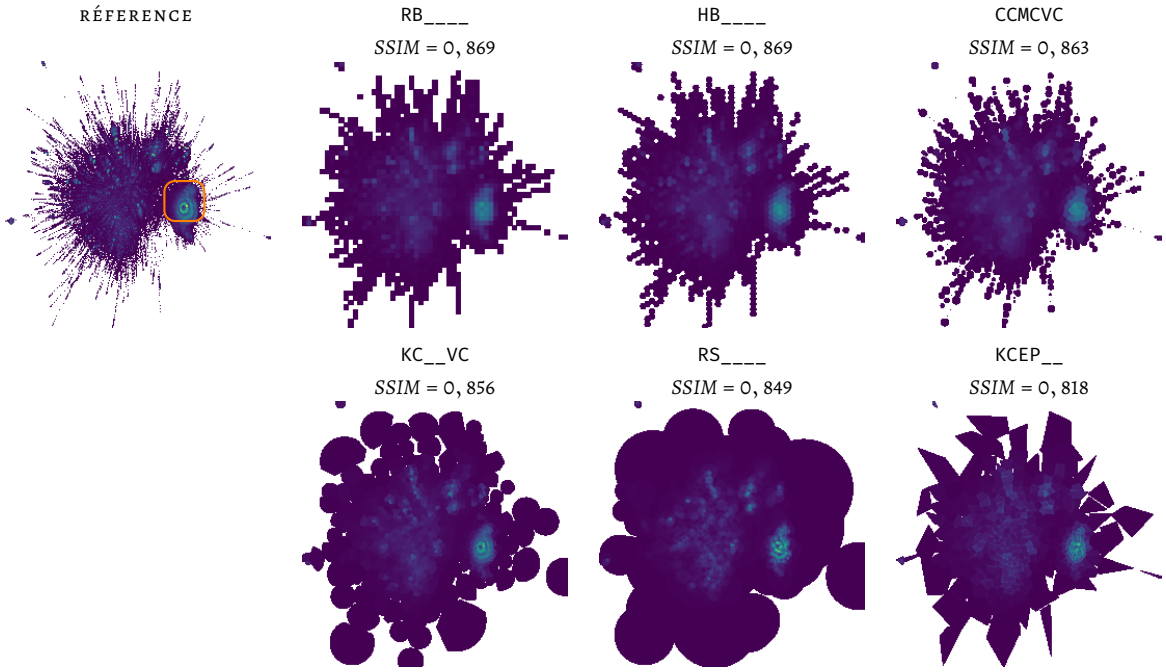


FIGURE 4.18: Les scores de similarité SSIM évaluent uniformément l'ensemble de l'image alors qu'un jugement humain est susceptible d'accorder plus d'importance aux sous-parties remarquables comme des motifs particuliers. Les CDD présentées ici sont issues des positions nœuds d'un dessin du graphe `panama_papers` [132] pour $r = 5\,\%$. Leur qualité est faible selon SSIM (*poor*) avec les scores de la première ligne légèrement supérieurs à ceux de la deuxième. Sur cette première ligne, les contours des amas de points sont plus précis que sur la seconde, mais sur la seconde, le motif de cercles concentriques est retranscrit plus fidèlement que sur la première.

4.6.2 *Comparaison des stratégies par des utilisateurs*

Pour compléter les résultats de la comparaison par métrique, nous avons conduit une évaluation *subjective* de l'efficacité des stratégies qui s'appuie directement sur des jugements de perception humains. Comme la précédente, cette étude vise à comparer l'efficacité des stratégies entre elles, à budget égal. Comparé à l'évaluation objective, le périmètre d'évaluation est restreint aux trois taux de compression $r = 2\%$, $r = 10\%$ et $r = 30\%$ et aux six stratégies suivantes :

— 1er tiers : `RB____`, `HB____`, `CCMCVC`, `KC__VC` ;
— 2e tiers : `RS____` ;
— 3e tiers : `KCEP__`.

HYPOTHÈSES On s'attend à ce que les jugements humains soient généralement cohérents avec l'évaluation par métrique mais qu'ils privilégient la conservation de structures particulières de la densité telles que les motifs remarquables ou les contours des zones de densité. La figure 4.18 présente des CDD de différentes stratégies pour un jeu de données réel qui fournissent deux exemples. Les CDD de la première ligne (`RB___`, `HB___`, `CCMCVC`) retranscrivent plus fidèlement les bordures que celles de la seconde. Les stratégies de la seconde retranscrivent par contre plus fidèlement le motif de cercles concentriques sur cet exemple (cf. H_2). Nous avons donc les hypothèses suivantes :

H4 Les stratégies du 1er tiers surpassent les deux autres ;

H5 À bas budget, `KC__VC`, `KCEP__` et `RS____` ont de meilleurs résultats qu'à plus haut budget.

Données d'évaluation

L'étude comparative de l'efficacité des six stratégies est conduite selon un plan intra-sujet, c.-à-d. tous les participants sont soumis à toutes les conditions : stratégie, budget et jeu de

jeu de données	n	pixels d'intensité nulle %
sinusoid1	1.000.000	44,64
sinusoid0	1.000.000	46,42
sinusoid2	1.000.000	60,88
clusters1	1.000.000	61,77
linear0	1.000.000	61,85
linear1	1.000.000	62,5
linear2	1.000.000	67,65
clusters2	1.000.000	69,72
clusters0	1.000.000	69,78
panamapapers	743.254	75,25
brightkite	4.747.281	95,29

TABLE 4.9 – Caractéristiques des jeux de données utilisés et de leur images de référence. Les jeux de données sont ordonnés selon leur quantité de pixels vides sur l'image de référence (256 × 256). Les deux jeux de données réelles sont ceux en possédant la plus grande quantité.

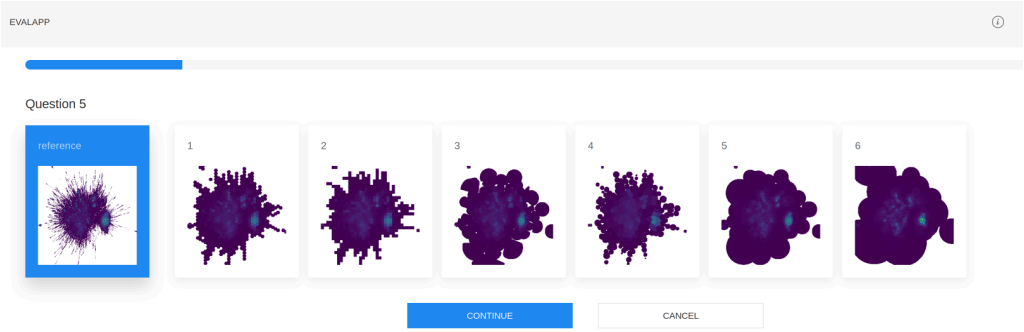


FIGURE 4.19 – Interface de l'expérience utilisateur. Chaque question affiche l'image de référence et les six images à ordonner. Les images à ordonner sont présentées dans un ordre aléatoire.

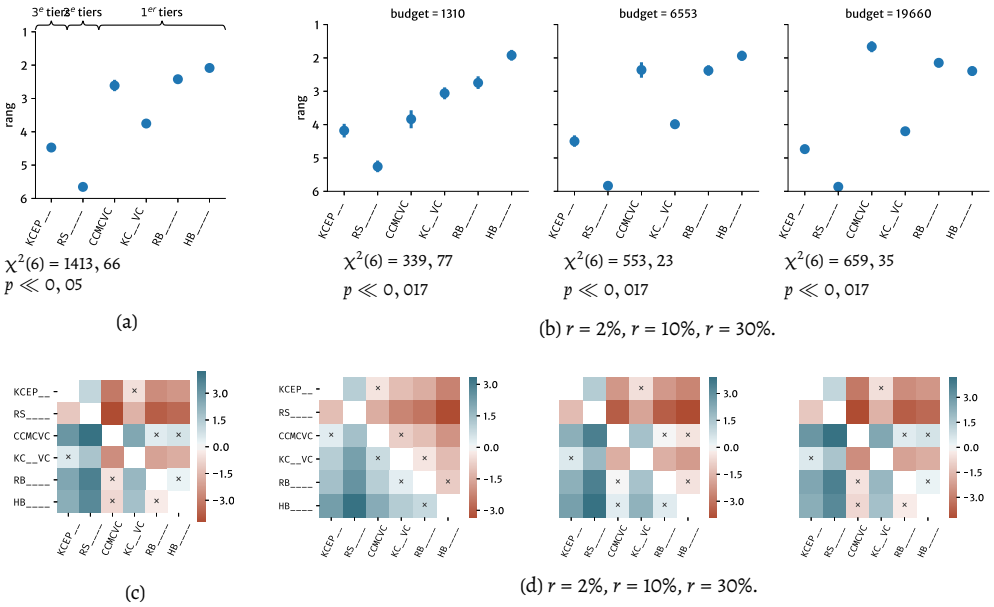
3 distributions	
3 instances	
9 jeux synthétiques	
+ 2 jeux réels	
11 jeux de données	
× 3 taux de compression	
33 questions	
× 25 participants	
495 classements	

TABLE 4.8: Conditions de l'expérimentation et nombre total de résultats (classements).

données (cf. tableau 4.8). Les jeux de données utilisés sont à la fois synthétiques et réels et sont listés sur le tableau 4.9. Les jeux de données synthétiques ($n = 10^6$) sont générés pour les trois distributions présentant les meilleurs résultats en moyenne dans l'évaluation précédente, soit : `linear`, `sinusoid` et `clusters`. Trois jeux de données sont générés pour chacune, et augmentés de deux jeux de données réelles : `panama_papers` (n =743 254) et `brightkite` (n =4 747 281).

PROCÉDURE ET PARTICIPANTS L'expérimentation est composée de deux phases : durant la première, les participants prennent connaissance de la consigne et traitent trois questions d'introduction dont les résultats ne sont pas pris en compte et dont l'ordre est identique pour tous les participants. Durant la seconde phase, les participants traitent les 33 questions (cf. tableau 4.8) qui sont présentées dans un ordre aléatoire. Comme pour l'étude précédente, les stratégies sont comparées à budget égal. Toutes les questions à traiter consistent à ordonner les images résultant des six stratégies, par une action de glisser-déposer, pour un même budget et un même jeu de données (cf. interface de l'expérimentation sur la figure 4.19). Puisqu'il est difficile de juger de la qualité d'une représentation sans point de comparaison, l'image de référence est présentée conjointement. Pour chaque question, les images à ordonner sont présentées dans un ordre aléatoire.

Les images sont rendues en utilisant une échelle de couleurs perceptuellement uniforme, du violet foncé pour les densités faibles au jaune pour les plus fortes, pour les pixels d'intensité non nulle. Les pixels d'intensité nulles sont représentés en blanc pour créer un fond contrasté



(a) (b) $r = 2\%$, $r = 10\%$, $r = 30\%$.

(c) (d) $r = 2\%$, $r = 10\%$, $r = 30\%$.

FIGURE 4.20 – Comparaison des six stratégies sur l'ensemble des jeux de données (25 participants). Rangée supérieure : classements moyens (barres d'erreur : intervalle de confiance à 95 %). Rangée inférieure : matrice des différences de classement avec significativité des différences (Conover).

similairement à celui d'un nuage de points. Cette échelle est celle utilisée pour toutes les images de CDD dans ce chapitre.

Nous avons recruté 25 participants (étudiants en master, doctorants, jeunes docteurs en informatique et informatique médicale) de notre université. L'étude est conduite sans limitation de temps. Les participants ont mis en moyenne 13,8 minutes à la terminer pour une moyenne de 25,4 secondes par question.

RÉSULTATS Les résultats de chaque question sont des classements qui ne suivent pas une distribution normale (test de Shapiro-Wilk, $\alpha = 0,05$). Un test de Friedman met en évidence les différences entre les classements des six stratégies sur l'ensemble des conditions ($\alpha = 0,05$) ainsi que pour les classements séparés par budget ($\alpha' = 0,05/6 = 0,0017$). La figure 4.20 rapporte les classements moyens des stratégies dans l'ensemble et par budget. Dans l'ensemble, les stratégies du 1er tiers sont significativement mieux classées que les deux autres. RS____ (2e tiers) est par contre significativement moins bien classée que KCEP__ (3e tiers). Au sein du 1er tiers, KC__VC est surpassée par les autres stratégies du groupe dans l'ensemble comme pour les deux budgets les plus élevés. Au budget le plus élevé, CCMCVC a un classement moyen supérieur à toutes les autres stratégies sans que cette tendance ne soit significative pour les stratégies de *binning*.

DISCUSSION L'hypothèse H_4 est vérifiée dans l'ensemble et pour tous les budgets. KC__VC par contre est classée à un rang significativement inférieur par rapport aux autres stratégies issues du 1er tiers et KCEP__ (3e tiers) présente des résultats significativement meilleurs que RS____ (2e tiers). Ces différences avec les résultats de l'analyse par métrique peuvent s'expliquer par la faible fidélité des contours produit par RS____ (et dans une moindre mesure KC__VC), illustrée par exemple sur la figure 4.18. Cette faiblesse est probablement accentuée pour les deux jeux réels qui présentent plus de points isolés et périphériques que les jeux de données synthétiques. À bas budget ($r = 2\%$), KC__VC est mieux classée en moyenne qu'aux deux autres budgets, mais ce n'est pas le cas de RS____ et KCEP__ ce qui ne supporte que partiellement l'hypothèse H_5.

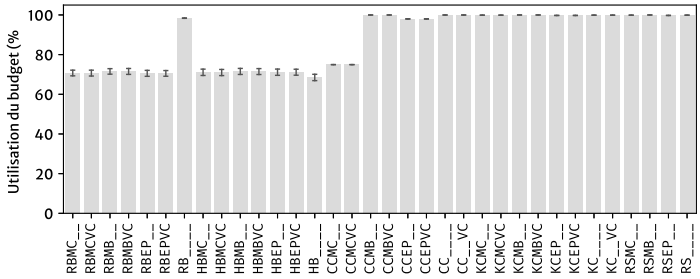


FIGURE 4.21 – Utilisation moyenne du budget alloué ($taille(\tilde{G})/k$) par chaque stratégie dans l'expérience.

4.7 Discussion

Nos résultats expérimentaux montrent que, pour les stratégies testées, les résultats d'évaluation par métrique et humain sont cohérents en général. Les trois stratégies présentant les meilleurs résultats dans l'évaluation subjective et dans l'évaluation objective sont `CCMCVC`, et les deux stratégies de *binning* standard `RB____` et `HB____`.

Stockage des géométries

Les stratégies évaluées sont issues de la composition de méthodes de partitionnement standards et de formes géométriques utilisées dans ces précédents travaux [84, 103, 171]. Nous avons comparé ces stratégies à budget égal où le budget correspond à la taille de stockage d'une géométrie compressée. Cette approche a deux principales limitations.

La capacité des stratégies à utiliser pleinement leur budget, c.-à-d. avoir $k = taille(\tilde{G})$, diffère suivant les méthodes de partitionnement utilisées et dépend dans notre étude des valeurs de budgets choisies. Certains partitionnements atteignent systématiquement exactement m parties alors que ce n'est pas le cas pour d'autres comme le partitionnement par canopées (cf. ci-dessus) et les *binning*. Pour les partitionnements par *binning*, le nombre effectif de cellules est le carré le plus proche à l'inférieur de m. Par conséquent, la probabilité d'obtenir exactement m cellules décroît lorsque que m augmente. Dans nos expériences, certaines stratégies n'utilisent qu'environ 70 % de leur budget alloué en moyenne alors que les stratégies basées sur l'échantillonnage aléatoire et les k-moyennes l'utilise presque totalement en moyenne (cf. figure 4.21).

L'encodage des géométries est simple et non compressé. D'autres encodages de la géométrie compressée ou la considération de sa taille compressée (sans pertes) pourrait résulter en (1) une meilleure qualité pour un budget moindre, et (2) un changement dans la comparaison des stratégies. Notre encodage a toutefois le mérite d'être déterministe : il peut répondre à une garantie *en taille*. D'autre part, il semble plus compact que des formats standards d'encodage de géométries, y compris le format `svgz` qui est compressé. La tableau 4.10 compare les tailles de plusieurs images géométriques élémentaires dans notre format (avec 8 octets par valeur flottante) et dans le format `SVG`, `SVGZ` (compressé) et `GeoJSON`. Ces résultats ne sont pas surprenants puisque notre format n'est pas générique et donc plus compact, mais aussi ici évalué en stockage binaire et non textuel.

Enfin, notre implémentation utilise un stockage *complet* pour le *binning*, c.-à-d. qui stocke la valeur de chaque *bin*, alors que dans certains cas le stockage *creux* n'incluant que les *bins* non-nuls est plus compact [107].

Périmètre des stratégies et comparaisons

Nous avons concentré notre étude sur les stratégies d'abstraction pour les cartes de densités discrètes, non-lissées. Pour limiter le nombre de comparaisons, les études ne considère que les

GÉOMÉTRIE	MESURÉ GÉOMÉTRIE (OTETS)				
	# valeurs	notre format	SVG	SVGZ	GeoJSON
1 cercle	2	8	972	549	3,2K
1 rectangle aligné	4	16	975	554	277
1 heptagone (7 côtés)	14	56	874	527	325
1 hendécagone (11 côtés)	22	88	1,1K	632	565
ensemble des 4 formes	42	168	2,2K	955	2,9K
400 cercles	800	3,2K	137K	11K	331K
700 cercles	1,4K	5,6K	250K	20K	586K
1K cercles	2K	8K	338K	24K	823K
2K cercles	4K	1,6K	674K	51K	1,1M

TABLE 4.10 – Comparaison de coûts de stockage de plusieurs géométries (non pondérées) pour des formats vectoriel standard et notre format. Le format SVG a un coût très important qui est cependant atténué par sa compression efficace (SVGZ). Le format GeoJSON a le désavantage de ne pas permettre directement le stockage de cercles. Notre format, ici évalué pour des flottants sur 8 octets (64 bits) est de façon non surprenant plus compact que ces formats standards.

quadrilatères pour les stratégies codées EP, et n'évalue donc pas les performances des pentagones, heptagones, etc. Ce choix a été orienté par l'implémentation : les polygones englobants utilisés étant à la fois non-minimaux et convexes, il était peu probable que des polygones d'ordre plus élevé présentent un avantage aux budgets utilisés.

Les résultats des études pourraient être complétés par la comparaison à d'autres techniques existantes comme le *binning adaptatif* [140] qui forme des *bins* de valeur égale plutôt que de surface égale. Elle permet aussi d'envisager de nouvelles techniques. Nous sommes concentrés sur les géométries n'utilisant qu'un type de forme (à l'exception des cas dégénérés). En particulier, dans notre implémentation des p-gones, les p points ne sont pas tous nécessairement utilisés et le budget correspondant est perdu. Une utilisation du budget de manière plus flexible entre les polygones de chaque partie de manière à faire dépendre le nombre de points de chaque polygone de la complexité de l'enveloppe convexe ou concave du groupe de points pourrait être plus efficace.

Comparaison par métriques et utilisateur

La comparaison des stratégies par métrique s'applique au niveau des cartes de densité discrète (CDD), c'est-à-dire les cartes de densité discrétisées spatialement mais pas en valeur. Elle a l'avantage d'isoler le problème de la reconstruction de la carte de densité, de celui de l'encodage des valeurs de densité en couleurs pour la présentation à l'utilisateur. Ainsi, elle permet d'évaluer la compression géométrique en écartant le biais qu'introduisent les choix d'encodage en couleurs [128], alors que l'évaluation utilisateur est nécessairement dépendante de ces choix. L'évaluation par métrique a aussi l'avantage de permettre la comparaison de nombreuses stratégies à la fois rapidement. L'évaluation utilisateur apporte un complément important pour mettre ces résultats en perspective. Les performances de la stratégie RS____ sont par exemple différentes dans l'évaluation utilisateur mais cela pourrait être dû au biais de l'échelle de couleurs utilisée qui introduit un contraste important entre les valeurs de densité exactement nulle (blanc) et proche de zéro (violet foncé).

Coût de compression

L'objectif des stratégies de compression géométrique présentées est de permettre l'exploration efficace de grands nuages de points en constituant des hiérarchies d'information géométrique précalculées. La compression géométrique est ici utilisée pour réduire l'espace de stockage nécessaire

relativement au stockage d'une pyramide d'image et s'ajuster à la définition de la vue présentée à l'utilisateur.

Dans un contexte de mise à jour fréquente, les stratégies doivent également être comparées en termes de complexité et d'implémentation dans un environnement distribué, en plus de leur efficacité. Les techniques de partitionnement par exemple n'ont pas toutes le même coût. Le partitionnement par canopées est plus efficace calculatoirement que les k-moyennes avec une complexité en $O(n \cdot log(k))$ si les k représentants sont stockés dans une structure appropriée. Les k-moyennes effectuent kn comparaisons au pire à chaque itération, soit une complexité au pire $O(tkn)$ avec t le nombre d'itérations. Cependant, le partitionnement par canopées ne prend pas comme valeur d'entrée le nombre de représentants k mais une distance d'agrégation d. En conséquence, le nombre de parties dépend à la fois de d, de la distribution spatiale des points et de l'ordre de parcours. Pour obtenir le partitionnement, nous recherchons itérativement la distance d'agrégation d qui résulte en un nombre de parties proche de m en utilisant un algorithme de recherche de racine. Les trois autres partitionnements ont l'avantage d'être bien plus efficace car non itératifs.

4.8 Conclusion

Nous avons présenté une approche de compression géométrique pour des ensembles de points. La géométrie compressée est destinée à supporter une représentation visuelle d'ensemble de points par carte de densité. La compression de la géométrie est une alternative au transfert d'images (compressées ou non) dans un contexte de visualisation déportée qui apporte plus d'interactivité sur le client de visualisation (sélection, redimensionnement, changement d'encodage des couleurs). Pour atteindre des taux de compression élevés et tirer parti des limitations en détails lié à la perception humaine et à la résolution des écrans, nous nous sommes intéressées à la compression avec pertes qui conserve deux propriétés de la carte de densité de référence : sa masse et sa couverture sur l'image. Le processus de rendu conservateur implémenté pour ces géométries compressées permet de garantir la conservation de ces deux propriétés après discrétisation spatiale.

Nous avons proposé une méthode générale de composition de stratégies de compression géométrique pour des ensembles de points qui respecte les deux propriétés de conservation énoncées et utilise des techniques existantes de partitionnement de données et de représentation de groupes sur les nuages de points. Nous avons ensuite implémenté 34 de ces stratégies par composition et les avons comparés pour plusieurs taux de compression élevés relativement à la taille des données ($\tau > 99\%$) dans deux études, l'une utilisant une métrique de similarité et l'autre utilisant des jugements humains.

La comparaison par métrique a permis de comparer toutes les stratégies et d'isoler six stratégies surpassant les autres en moyenne. Ces stratégies ont le point commun de résulter en des formes géométriques sans superposition et à faible coût par forme. On retrouve parmi elles, les deux approches usuelles par *binning* (rectangulaire ou hexagonal) et cinq autres stratégies utilisant les partitionnements par k-moyennes ou canopées couplé au rognage Voronoï (`KCMCVC`, `CCMCVC`, `KCMBVC`, `CC__VC` et `KC__VC`). Parmi ces dernières `CC__VC` est celle présentant les meilleurs résultats.

La comparaison par des utilisateurs a servi à comparer six stratégies choisies à la fois parmi les meilleures, les pires et les moyennes telles que catégorisée lors la première étude. Les résultats sont globalement cohérents avec ceux de l'étude précédente et montre que la stratégie `CCMCVC` n'est pas significativement différentes des *binning* selon des jugements humains.

Ces résultats sont importants car la régularité de l'abstraction par *binning* présente des limites pour la retranscription de détails fins à bas budget (cf. motifs concentriques de la figure 4.18). Or, la retranscription de détails particuliers, même à niveau d'agrégation élevé peut être décisive pour l'orientation de l'exploration utilisateur, notamment sur la vue initiale. Il est donc important d'identifier des stratégies d'abstraction alternatives, c.-à-d. aux performances comparables, mais

non-régulière pour envisager de contourner cette limite du *binning*. Nos deux études suggèrent respectivement que CC__VC et CCMCVC constituent de telles alternatives pour les budgets considérés. Des études complémentaires futures pourraient consolider ces résultats en comparant sur un plus grand nombre de jeux de données réelles l'ensemble des stratégies du 1er tiers entre elles, notamment CC__VC et CCMCVC.

5 HiePaCo : Coordonnées parallèles hiérarchiques abstraites

Dans les précédents chapitres, nous avons d'une part répondu à la problématique de scalabilité *visuelle* dans les vues multiples, et d'autre part étudié une solution de compression de données géométriques dans un contexte de visualisation déportée servant à la scalabilité *computationnelle* pour les cartes de densité. Dans ce chapitre nous traiterons de données multi-dimensionnelles et nous nous intéressons aux coordonnées parallèles, déjà présentées en section 2.1, page 5, comme moyen pour les représenter. Plus particulièrement, la problématique est d'examiner les techniques permettant à la fois la scalabilité visuelle et la scalabilité computationnelle d'un système interactif d'exploration de données multi-dimensionnelles. Comme dans le chapitre précédent, on s'intéressera à l'utilisation d'une infrastructure distribuée distante et ainsi, un point essentiel de la conception du système et de ses interactions sera le respect d'un budget pour la quantité de données transférées sur le réseau.

Nous présentons dans un premier temps un formalisme pour les coordonnées parallèles abstraites, puis un système permettant la sélection en temps interactif sur une représentation agrégée de plus d'un milliard d'entités et enfin HiePaCo [1], une version du système permettant également de naviguer entre plusieurs niveaux d'agrégation des données jusqu'à la granularité des entités.

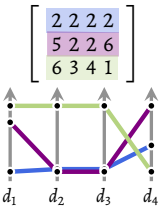


FIGURE 5.1: Chaque entité est une polyligne croisant les quatre axes.

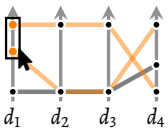


FIGURE 5.2: Surbrillance en orange d'entités sélectionnées par brossage de l'axe pour d_1.

5.1 Problématique et existant

Dans un système de coordonnées parallèles [82], les m dimensions des données sont associées à m axes, usuellement placés parallèlement. Une entité est représentée par une ligne brisée (polyligne) croisant ces m axes en sa valeur pour la dimension correspondante comme illustré sur la figure 5.1. Cette disposition possède de nombreux avantages pour la visualisation de données multi-dimensionnelles, notamment : la représentation de plus de trois dimensions à la fois (jusqu'à une douzaine), et le traitement uniforme de ces dimensions.

L'interaction de sélection d'entités, en général sous forme de brossage et lien, est cruciale dans les coordonnées parallèles. Elle permet de *tracer* une entité ou un sous-ensemble des entités en levant l'ambiguïté des croisements aux intersections sur les axes et permet de comparer certaines entités à la tendance générale ainsi que d'identifier les entités aberrantes ou isolées. La sélection est en général implémentée par brossage d'un axe et révélée par la mise en surbrillance des polylignes concernées avec une couleur distinctive, comme illustré sur la figure 5.2.

L'identification de relations de corrélation (respectivement anti-corrélation) entre deux dimensions correspondent, sur les coordonnées parallèles, à une recherche de motif spécifique : l'absence d'intersection entre les lignes (respectivement une forte densité d'intersections au même point). Puisqu'une représentation *conventionnelle* de coordonnées parallèles affiche un axe par dimension, seule une fraction des paires de dimensions est représentée et plus précisément, $m-1$ des $\frac{m(m-1)}{2}$ paires de dimensions pour m dimensions. Une manière de permettre l'analyse des relations entre les autres paires de dimensions, est de proposer le ré-ordonnement interactif des axes. Le couple $\{d_1, d_4\}$ par exemple n'est pas représenté sur la figure 5.1 et l'est après interversion des axes d_1 et d_4 sur la figure 5.3. L'autre manière d'accéder à toutes les couples de dimensions, c.-à-d. tous les *sous-espaces 2D* [2], est de dé-multiplier les axes [104, 77].

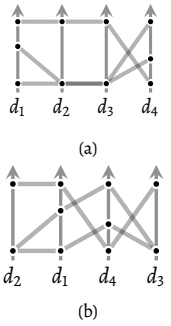


FIGURE 5.3: Les quatre axes ne permettent pas de représenter tous les sous-espaces 2D à la fois. (a) Vue initiale. (b) Vue après déplacement des axes d_1 et d_2, et d_3 et d_4.

1. pour **Hie**rchical **Pa**rallel **Co**ordinates
2. Nous nous intéressons ici seulement aux sous-espaces alignés aux dimensions du jeu de données.

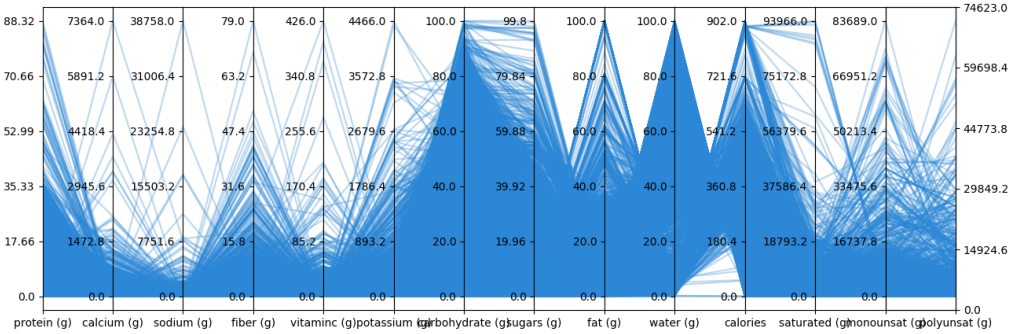


FIGURE 5.4 – Encombrement visuel dans les coordonnées parallèles (238 mille entités).

(a) Courbes de Bézier [81].

(b) Faisceautage des lignes [178].

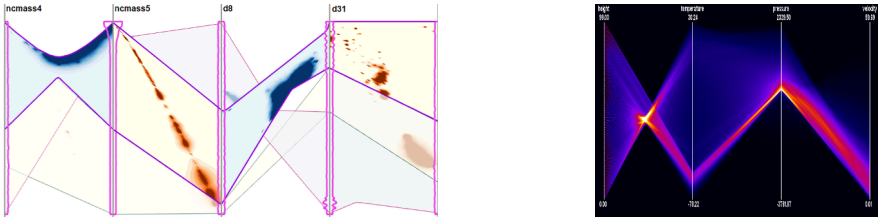
FIGURE 5.5 – Approches géométriques pour la réduction de l'encombrement visuel utilise des courbes, faisceau de courbes ou déplacement des sommets des lignes pour faciliter l'interprétation ou réduire les chevauchements de lignes.

5.1.1 *Scalabilité visuelle*

Une des limites principales des coordonnées parallèles conventionnelles est la superposition des lignes qui crée rapidement de l'encombrement visuel lorsque le nombre d'entités augmente comme illustré sur la figure 5.4 présentant 238 mille entités. L'encombrement visuel résulte de la multiplication des croisements et chevauchement entre lignes qui peut masquer les motifs caractéristiques de la structure des données. De part sa géométrie, les coordonnées parallèles sont particulièrement sujettes à ce phénomène puisque l'entité graphique d'une entité des données, la ligne, couvre de nombreux pixels même lorsqu'elle est dessinée la plus fine possible. HEINRICH et WEISKOPF [79] distinguent trois types de techniques visant à pallier l'encombrement visuel dans des coordonnées parallèles : le filtrage incluant l'échantillonnage [51] et les interactions de mise en surbrillance ; l'agrégation s'appuyant sur des agrégats visuels pour limiter le nombre d'entités dessinées ; et enfin la déformation spatiale dont l'objectif est de diminuer les superpositions en déplaçant ou déformant les polylignes pour faire meilleur usage de l'espace écran. Nous écartons dans cet état de l'art les techniques de filtrage et échantillonnage et subdivisons les approches représentant toutes les données en deux catégories : les approches géométriques qui altèrent le dessin des polylignes et les approches par agrégation qui résument la vue en représentant des groupes d'entités ou des champs scalaire de densité d'entités.

Approches géométriques

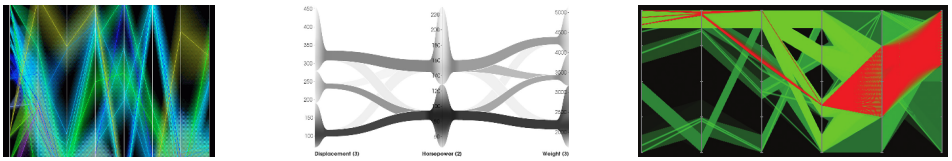
Les approches géométriques représentent une entité visuelle par entité des données. Certaines méthodes utilisent des courbes plutôt que des lignes brisées pour représenter les entités d'une manière moins ambiguë aux intersections avec les axes, pour réduire les superpositions ou encore accentuer des groupements d'entités [68, 81, 178, 114, 162]. D'autres techniques déforment les intersections sur les axes pour clarifier la vue [125, 68]. Plusieurs de ces techniques sont représentées sur la figure 5.5. Le passage à l'échelle de ces approches est fondamentalement limité par le fait qu'elles dessinent un élément visuel par entité et ainsi restent limitée par l'espace



(a) DSPCP [124].

(b) Coordonnées parallèles continues [78].

FIGURE 5.6 – Approches par carte de densité. Ces approches représentent une carte de densité des données et élimine ainsi la problématique de superposition de lignes.



(a) Partitionnement multi-dimensionnelles [62].

(b) Partitionnement par dimension [129].

(c) Histogramme (sauf entités isolées et sélectionnées) [127].

FIGURE 5.7 – Exemples d'approches par partitionnement multi-dimensionnel (a) et dimensionnel (b-c).

écran disponible. Les deux catégories suivantes pallient cette limite au prix d'une réduction et donc perte de l'information.

Approche par l'agrégation

Les approches *densité*, transforment les données en une fonction de densité [78, 124, 89]. Par définition ces approches ne souffrent pas de la superposition des entités géométriques puisqu'elles transforment les données en champ scalaire (cf. figure 5.6). Elles sont généralement lissées et dirigent donc l'attention de l'utilisateur premièrement vers les tendances globales des données au détriment de l'observation de phénomènes concernant un ou peu d'entités des données.

Les approches par *partitionnement* agrègent les entités suivant une partition et représentent ces agrégats par leur enveloppe (extrema) ou encore en leur attribuant une taille d'autant plus grande qu'ils représentent d'entités (poids). On distingue deux types d'agrégation : l'agrégation des entités multi-dimensionnelles [62, 114] et l'agrégation des valeurs par dimension [129, 25, 127, 65]. Parmi les représentations utilisant l'agrégation par dimension, certaines utilisent un partitionnement issu d'un attribut catégoriel des données [96], d'autres d'un algorithme de partitionnement des valeurs [129] ou encore d'un histogramme, c.-à-d. *binning* des données [127, 65].

De manière générale, les approches par partitionnement fournissent une vue d'ensemble efficace des données et sont visuellement scalables relativement au nombre d'entités puisqu'elles ramènent le problème de représentation d'un grand jeu de données à celui d'un petit jeu de données. Cependant, des interactions sont nécessaires pour ajuster le niveau de détail de l'abstraction, en particulier pour accéder à de l'information au niveau des entités (niveau le plus fin). Les travaux existant concernant ces interactions sont détaillés plus loin, en section 5.4, page 90.

5.1.2 *Scalabilité computationnelle*

Parmi les verrous de la scalabilité computationnelle des coordonnées parallèles, on peut distinguer trois facteurs : la complexité de rendu qui dépend du nombre d'entités dessinées et de leur superposition, le coût de parcours des entités, et enfin le coût du transfert réseau induit par chaque interaction dans le contexte de visualisation déportée. La complexité des interactions est en général linéaire avec la taille des données lorsque l'on considère les partitionnements

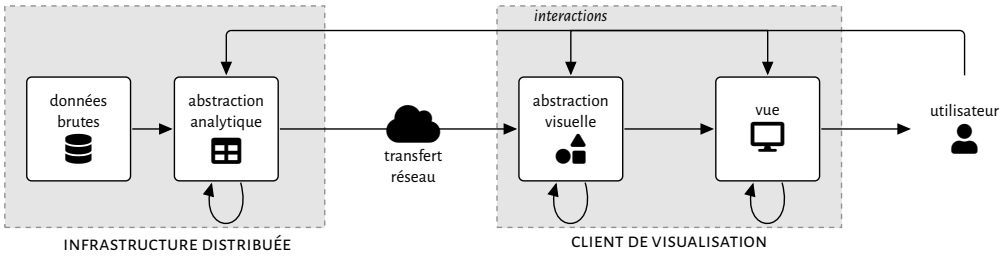


FIGURE 5.8 – Pipeline de visualisation déportée s'appuyant sur une plateforme distribuée.

pré-calculés puisque les traitements consistent essentiellement en du filtrage et des calculs de statistiques (moyenne, somme, etc).

Les techniques utilisées pour le passage à l'échelle des traitements sont l'utilisation de processeurs graphiques modernes combinée à l'agrégation [62, 89, 12], le parallélisme sur supercalculateur [140] et le pré-calcul [124, 140]. Pour être interactif un système doit opérer en dessous de la seconde pour les opérations telles que la sélection et en dessous de *100 ms* pour les animations [27]. Atteindre ces performances est en général un compromis entre le prix du matériel utilisé, du pré-calcul et de l'approximation. RÜBEL et al. [140] par exemple utilisent un supercalculateur et calcule une sélection de 500 entités sur un jeu de données de 200 millions d'entités en *0,15 s* avec 100 unités de calculs. Notre objectif est d'atteindre des performances satisfaisantes en utilisant une infrastructure distribuée utilisant du matériel moins onéreux. Les approches de scalabilité par l'abstraction sont susceptibles de contribuer à la diminution du temps de rendu et de transfert entre un client et son serveur étant donné qu'elles dessinent moins d'entités et à la diminution des temps de traitement lorsque les interactions utilisent du pré-calcul [124].

Dans la suite de ce chapitre nous présentons un formalisme pour la structure des coordonnées parallèles conventionnelles et agrégées (ou *abstraites*) et des implémentations de système de visualisation interactive déportée pour ce type de représentations utilisant une infrastructure distribuée.

5.2 Formalisme et environnement d'implémentation

Le formalisme présenté permet d'exprimer les modifications induites par les interactions de sélection, déplacement d'axes et de navigation hiérarchiques sur la structure de la représentation. Il constitue donc un socle de réflexion pour la conception d'interactions validant des contraintes issues de l'environnement d'implémentation tel qu'un budget en entités visuelles. Cet environnement d'implémentation est celui de la visualisation déportée où les traitements des données complètes sont effectués sur une infrastructure distribuée et la visualisation des données agrégées sur un client léger tel qu'illustré sur la figure 5.8.

5.2.1 *Définitions et notations*

Ce qui suit introduit les termes utilisés par la suite dans cette section pour décrire les matrices, graphes, partitions et hiérarchies.

Matrice

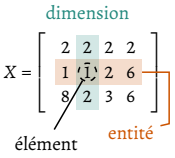


FIGURE 5.9: Matrice pour trois entités et quatre dimensions.

On modélise des données multi-dimensionnelles à n entités et m dimensions par une matrice de scalaires $X \in \mathbb{R}^{n \times m}$ où les lignes de X correspondent aux entités et ses colonnes aux dimensions. On note $x_{ij} \in \mathbb{R}$ les éléments de X, et x_{i*} les entités de X avec $i \in \{1, \ldots, n\}$ et $j \in \{1, \ldots, m\}$.

On désignera par d_i les m sous-espaces 1D des données et par $\{d_i, d_j\}$ avec $i \neq j$ les $\frac{1}{2} \cdot m(m-1)$ sous-espaces 2D des données.

Graphe

Un *graphe* est une structure définie par un ensemble de *nœuds* usuellement noté V (pour *vertices*) et un ensemble de *paires* de nœuds, appelées *arêtes*, noté E (pour *edges*) et tel que $E \subset \{\{u, v\},\ u \in V, v \in V\}$. Un graphe est usuellement noté $G = (V,\ E)$. Plus précisément, ce type de graphe est un graphe *simple* et *non-orienté*, c'est-à-dire qu'il n'existe qu'une ou aucune arête entre deux nœuds du graphe. Par la suite tous les graphes seront considérés simples et non-orientés. On dit que les deux nœuds formant une arête sont *adjacents* ou *connectés*. Les deux nœuds définissant une arête sont appelés les *extrémités* de cette arête. On appelle *ordre* d'un graphe son nombre de nœuds $|V|$. On appelle *voisins* d'un nœud du graphe l'ensemble des nœuds qui lui sont connectés.

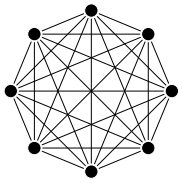


FIGURE 5.10: Graphe K_8 : $|V| = 8$, $|E| = 8 \times 7 = 56$.

Le *sous-graphe* d'un graphe $G = (V,\ E)$ induit par $V' \subset V$ est le graphe défini par les nœuds V' et par les arêtes de E dont les extrémités sont dans V'. Ce sous-graphe est noté $G[V']$. De même, le *sous-graphe* d'un graphe $G = (V,\ E)$ induit par $E' \subset E$ est le graphe défini par les nœuds de V étant extrémités des arêtes de E'. Ce sous-graphe est noté $G[E']$.

Une *chaîne* dans un graphe $G = (V,\ E)$ est une suite finie de nœuds de V telle qu'il existe une arête dans E connectant deux nœuds consécutifs de la suite. Une *chaîne* est *élémentaire* si elle ne contient aucune répétition de nœud. Dans la suite, une chaîne sera élémentaire. Un graphe est *connexe* s'il existe une chaîne connectant toute paire de ses nœuds. Un graphe non-connexe peut être décomposé en *composantes connexes* qui sont les sous-graphes connexes maximaux du graphe.

Un graphe *complet* est un graphe dont tous les nœuds sont adjacents deux à deux. On nomme K_n le graphe complet d'ordre n (unique à un isomorphisme près). Généralement, on représente un graphe simple non-orienté par un diagramme nœuds-liens comme sur la figure 5.10 pour le graphe K_8. Une *clique* d'un graphe est un sous-graphe complet du graphe.

Partition et graphe quotient

Une *partie* d'un ensemble A est un sous-ensemble de A non-vide. Une *partition* P d'un ensemble A est un ensemble de parties de A, deux à deux disjointes et dont l'union est A soit $P = \{P_i \subset A, i \in \{1, \dots, k\}\}$ où

— $A = \bigcup_{1 \leqslant i \leqslant k} P_i$;

— $\forall i,\ P_i \neq \emptyset$;

— $\forall i \neq j,\ P_i \cap P_j = \emptyset$.

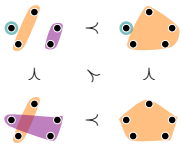


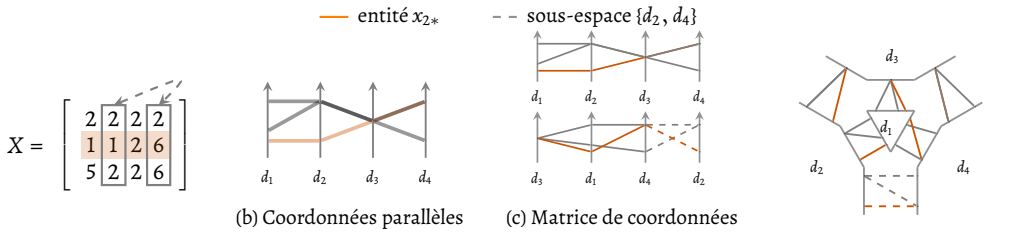
FIGURE 5.11: Relation d'ordre sur les partitions d'un ensemble à 5 entités ($|A| = 5$).

Pour deux partitions P et R d'un même ensemble A, on dit que R raffine P si et seulement si chaque partie de R est un sous-ensemble d'une partie de P. Vu autrement, R est un re-partitionnement de P. Cette relation est notée $R \prec P$ et correspond à un ordre partiel sur les partitions d'un même ensemble. On dit alors que R est plus *fine* que P ou que P est plus *grossière* que R. La partition la plus fine de A est celle constituée d'un singleton par élément de A. Le nombre de partitions différentes d'un même ensemble de taille n est exponentiel et donné par le nombre de Bell B_n dont les premiers termes sont 1, 1, 2, 5, 15, 52, 203, 877, 4140 et 21147. Ses termes sont donnés par la relation récurrente : $B_n = \sum_0^{n-1} B_k \binom{n-1}{k}$ où $\binom{a}{b}$ est le coefficient binomial.

Une partition $P = \{P_1, \dots, P_k\}$ d'un ensemble ordonné $A = \{a_1, \dots, a_n\}$ est une partition *intervalle* si et seulement si $\forall i < j,\ \forall a \in P_i,\ b \in P_j,\ a < b$. Il y a 2^{n-1} partitions de la sorte. On se limitera à ce type de partition pour les ensembles de scalaires (partitionnement de valeurs dimensionnelles).

Un graphe *k-parti* est un graphe pour lequel il existe une partition des nœuds en k parties telle que chaque partie est un *stable* du graphe, c'est-à-dire un ensemble de nœuds deux à deux non adjacents.

Étant donné une partition P des nœuds d'un graphe $G = (V,\ E)$, le graphe *quotient* de G relativement à P a pour nœuds chaque partie de P et connecte deux parties P_i et P_j pour $i \neq j$



(a) Matrice X. (b) Coordonnées parallèles conventionnelles. (c) Matrice de coordonnées parallèles [77]. (d) Many-to-many plot [104].

FIGURE 5.14 – Différentes dispositions pour les coordonnées parallèles pour X : (b) coordonnées parallèles conventionnelles utilisant une couleur semi-transparente, (c-d) techniques issues des coordonnées parallèles représentant tous les sous-espaces 2D.

si et seulement s'il existe au moins deux nœuds $u \in P_i$ et $v \in P_j$ tels que $\{u, v\} \in E$. Le graphe quotient de G relativement à P est noté $G_{/P}$. On appelle *méta-nœuds* et *méta-arêtes* les nœuds et arêtes, respectivement, d'un graphe quotient. Par extension, un graphe quotient $G_{/P}$ est plus grossier (respectivement plus fin) qu'un autre graphe quotient $G_{/R}$ si $P \succ R$ (respectivement si $P \prec R$).

Arbre et coupures

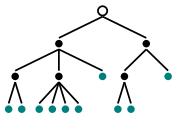
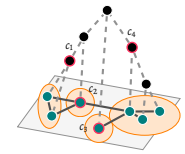

racine
feuille
nœud interne

FIGURE 5.12: Arbre enraciné d'arité 4 et de hauteur 3.



graphe
arbre de partition
coupure $c_1, \ldots, c_4$
partition de G issue de la coupure.

FIGURE 5.13: Une coupure de l'arbre de partitionnement d'un graphe correspond à une partition des nœuds du graphe.

Un *arbre* est un graphe non orienté, connexe et acyclique, noté $T = (V, E)$. Deux nœuds dans un arbre ne sont connectés que par une unique chaîne élémentaire. Un arbre est *enraciné* lorsqu'un de ses nœuds est étiqueté comme la *racine*. Par la suite on ne considérera que des arbres enracinés. On utilisera de manière interchangeable les termes *arbre* et *hiérarchie*. La *profondeur* (aussi *niveau*) d'un nœud de l'arbre est sa distance à la racine. La *hauteur* de l'arbre est la profondeur maximale dans cet arbre. L'arbre de la figure 5.12 par exemple est de hauteur 3.

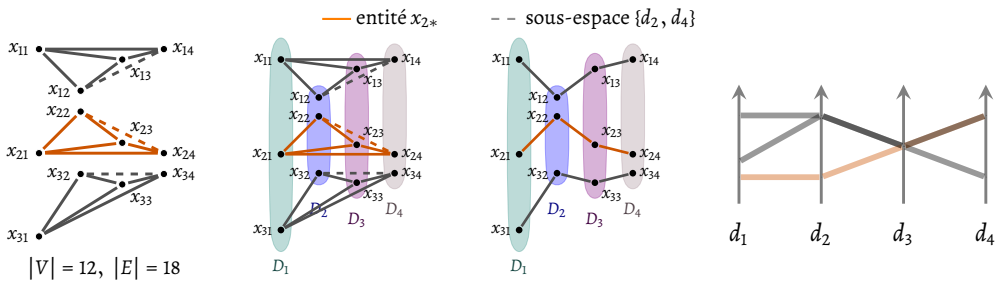
Pour tout nœud $v \in V$ différent de la racine, on distingue deux types de voisins : les voisins de v de profondeur supérieure à celle de v, appelés ses *enfants*, et celui de profondeur inférieure, appelé son *parent*. Les nœuds de la chaîne connectant v à la racine sont appelés ses *ascendants*. L'ensemble des nœuds ayant v comme ascendants sont appelés les *descendants* de v. Les nœuds partageant le même parent sont dits *frères*. L'*arité* de l'arbre est le nombre maximal d'enfants d'un nœud. Les nœuds de l'arbre n'ayant aucun enfant sont les *feuilles* de l'arbre. Les nœuds n'étant pas des feuilles sont dits *internes* et chacun *couvre* un sous-ensemble des feuilles de l'arbre, plus précisément celles qui font partie de ses descendants.

Une *antichaîne* de l'arbre est un ensemble de nœuds $S \subset V$ tel qu'aucun nœud de S n'est ancêtre d'un autre nœud de S. Une antichaîne est dite *maximale* si elle ne peut être augmentée par aucun nœud de V sans être rendue caduque. On appellera *coupure* une antichaîne maximale. Les nœuds d'une coupure couvrent des sous-ensembles distincts des feuilles de l'arbre et les sous-ensembles couverts par les nœuds d'une coupure forment une partition des feuilles de l'arbre comme illustrée sur la figure 5.13. La relation d'ordre partiel existante entre les partitions d'un même ensemble existe également entre les coupures d'un même arbre. Une coupure composée des nœuds d'un même niveau est dite *équilibrée*. La coupure équilibrée de niveau i définit une partition plus grossière que celle définie par la coupure équilibrée composée des nœuds du niveau $j > i$.

Un *arbre de partition* d'un graphe G est un arbre T tel que les feuilles de T sont les nœuds de G. Ainsi les coupures de T définissent chacune une partition des nœuds de G et donc un quotient de G.

5.2.2 *Formalisme pour les coordonnées parallèles conventionnelles*

Comme mentionné précédemment, une représentation conventionnelle ne représente pas tous les sous-espaces des données. Sur la figure 5.14, les lignes en pointillés correspondent au sous-espace $\{d_2, d_4\}$ qui n'apparaît pas sur la représentation conventionnelle mais est visible sur les représentations matricielles [77] et l'arrangement *many-to-many* [104] respectivement sur les



(a) Graphe G_X. (b) Avec la partition D. (c) Sous-graphe. (d) Coordonnées parallèles.

FIGURE 5.15 – Du formalisme graphe à une représentation conventionnelle de coordonnées parallèles. Les figures s'appuient sur la matrice X précédemment définie. Le graphe G_X a trois composantes connexes qui sont isomorphes à K_6 (a) et est 4-parti (b). La structure de la représentation conventionnelle (d), est celle du sous-graphe $G_X[(d_1, d_2, d_3, d_4)]$ (c).

figures 5.14c et 5.14d. Sur les exemples présentés, la matrice de coordonnées parallèles et la vue *many-to-many* montrent les six paires de dimensions des données, soit 18 lignes.

On peut voir les polylignes d'une représentation conventionnelle de coordonnées parallèles comme les arêtes d'un graphe dont les nœuds seraient les sommets des polylignes. La figure 5.15a présente ce graphe pour la matrice X de taille 3×4 précédemment définie. Ce graphe possède 12 nœuds, un par élément de la matrice ; et 18 arêtes, 6 par entités pour les 6 paires de dimensions. Chaque ligne d'une vue de coordonnées parallèles correspond à une arête et chaque polyligne à une chaîne de ce graphe.

Dans le cas général d'une matrice $X \in \mathbb{R}^{n \times m}$, nous le notons $G_X = (V,\ E,\ w,\ D)$ où :

— V est l'ensemble des nœuds, correspondant aux éléments x_{ij} de X ;

— E l'ensemble des arêtes, connectant les nœuds appartenant à une même entité de X ;

— w une valuation $w : V \to \mathbb{R}$ associant aux nœuds leur élément de X ;

— D est une partition de V groupant les nœuds suivant leur dimension d'origine.

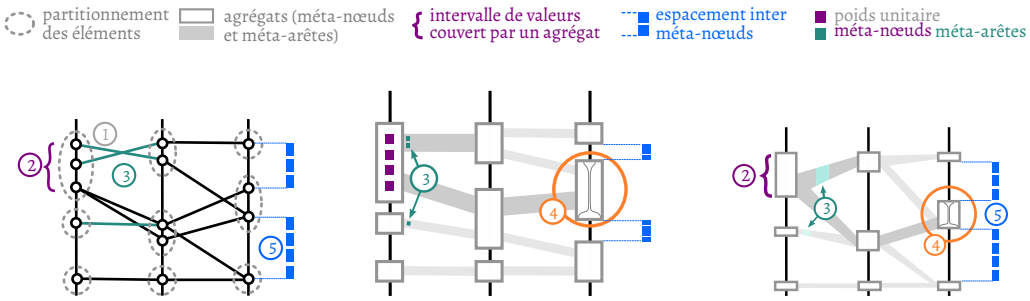
Les parties D_j de D sont étiquetées par la dimension à laquelle elles correspondent :

$$D_j = \{x_{ij},\ i \in \{1, \ldots, n\}\}.$$

Par construction, ce graphe possède n composantes connexes qui sont toutes des graphes complets d'ordre $m(m-1)/2$. Chaque composante possède un nœud dans chacune des m parties de D, ainsi ce graphe est également m-parti. Par exemple, le graphe de la figure 5.15b est 4-parti et ses trois composantes connexes sont d'ordre 6.

Les polylignes des coordonnées parallèles pour une séquence de dimensions $s = (d_i, \ldots, d_j)$ sont des chaînes du graphe et peuvent être isolées en s'intéressant au sous-graphe induit par l'ensemble des arêtes connectant des nœuds de deux dimensions consécutives dans la séquence. On appellera directement ce graphe le sous-graphe *induit par la séquence de dimensions*, que l'on notera $G_X[s]$. Dans notre exemple, la polyligne pour l'entité x_{2*} et la séquence de dimensions (d_1, d_2, d_3, d_4), en orange sur la figure 5.14b, correspond à la chaîne $(x_{21}, x_{22}, x_{23}, x_{24})$ sur le graphe. Le positionnement des nœuds sur les axes peut être déterminé par la partie de D à laquelle ils appartiennent (coordonnée x) et la valuation w (coordonnée y). Ainsi, un tel graphe contient une information équivalente à celle des données d'origine. Sa structure à proprement parler (V, E et D) ne dépend que de la taille des données c.-à-d. le nombre d'entités n et dimensions m.

Ce formalisme a l'avantage de présenter une structure proche de celle des représentations de coordonnées parallèles ce qui en fait une forme des données commode pour l'expression formelle d'interactions manipulant la structure de la représentation ainsi que pour l'évaluation des tailles de représentation en nombre d'éléments graphiques. En incluant toutes les paires de dimensions, il permet de modéliser d'autres dispositions que les coordonnées parallèles conventionnelles, y compris les deux dispositions de la figure 5.14 [104, 77].



(a) Encodage visuel conventionnel. (b) Encodage visuel *distribution*. (c) Encodage visuel *intervalle*.

FIGURE 5.16 – Principe des deux encodages visuels, *distribution* (b) et *intervalle* (c), proposés pour les coordonnées parallèles abstraites, ici illustrés pour le partitionnement défini par les lignes pointillées sur (a).

5.2.3 Environnement d'implémentation

Les systèmes de visualisation présentés dans ce chapitre s'appuient sur une plateforme distribuée composée d'une grappe d'une quinzaine de machines ayant chacune *64 Go* de mémoire vive et 2×6 cœurs (*2,10* GHz). Au sein de la plateforme, les liaisons réseaux ont une capacité de *1 Gbit/s*. La plateforme constitue un environnement Hadoop distribuant le stockage et le traitement des données. Nous évaluons les systèmes implémentés en termes de performances d'une part en mesurant les latences d'interaction, et en termes de scalabilité d'autre part, en évaluant l'évolution des performances en fonction des ressources utilisées.

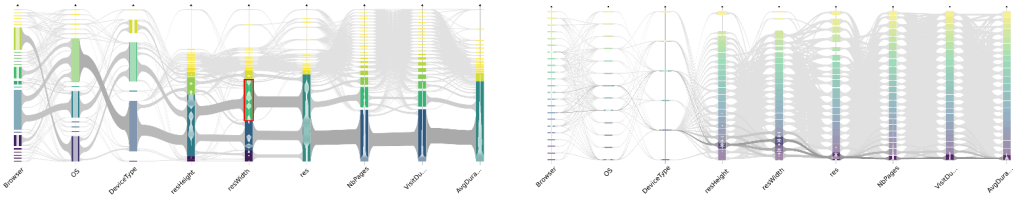
5.3 Coordonnées parallèles abstraites

Les problèmes de scalabilité visuelle évoqués en début de chapitre sont une motivation pour l'abstraction des coordonnées parallèles, c.-à-d. la représentation de données agrégées. Dans cette section, nous présentons d'abord une représentation visuelle de coordonnées parallèles abstraites basées sur des partitionnements 1D ainsi que ses interactions associées. Nous développons ensuite le formalisme présenté dans la section précédente pour formaliser cette agrégation et évaluer le coût du pré-calcul des interactions de sélection et ré-ordonnement. Enfin, nous présentons une implémentation et ses performances.

5.3.1 Encodage visuel & interactions

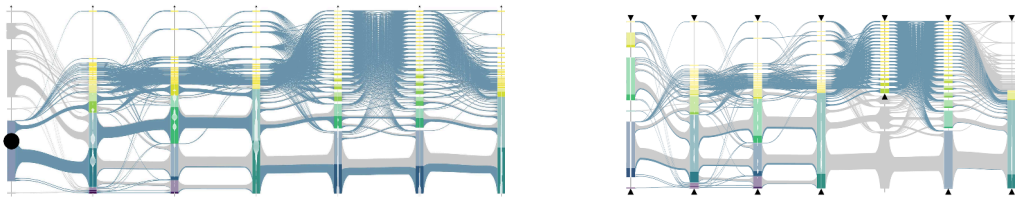
Nous proposons deux encodages visuels complémentaires et interchangeables interactivement pour des coordonnées parallèles agrégeant les valeurs dimensionnelles (partitionnement 1D). Un exemple de partitionnement 1D est présenté sur la figure 5.16a pour trois dimensions. La fusion des nœuds de chaque groupe conduit à la fusion de lignes entre elles sur les vues agrégées. Ces deux encodages visuels sont inspirés des Parallel Sets [96] (orienté *distribution*), et des coordonnées abstraites de PALMAS et al. [129] (orienté *intervalle*). Le premier utilise la hauteur ou épaisseur des méta-nœuds et méta-arêtes pour encoder la densité, c.-à-d. le nombre d'entités couvertes, alors que le second conserve la correspondance des positions aux valeurs des entités comme sur les coordonnées parallèles conventionnelles et utilise l'opacité pour transmettre la densité (cf. figure 5.16).

Les quatre entités encerclées et annotées ① sur la figure 5.16a forment, une fois agrégées, un méta-nœud possédant deux méta-arêtes sortantes. Dans l'encodage *distribution*, la hauteur des méta-nœuds et l'épaisseur des méta-arêtes indiquent leur poids ③. Dans l'encodage *intervalle*, la hauteur des méta-nœuds correspond à l'intervalle de valeurs qu'ils couvrent ②, l'opacité des méta-arêtes correspond à leur poids ③ et leur épaisseur est proportionnelle à la hauteur de leurs



(a) Encodage visuel *distribution*. (b) Encodage visuel *intervalle*.

FIGURE 5.17 – Application des deux encodages sur des données réelles. La différence d'encodage de la distribution est bien visible par la présence d'arêtes épaisses sur les valeurs basses de la plupart des dimensions sur a à laquelle correspond des arêtes plus foncées sur (b).



(a) Sélection *simple* d'un méta-nœud (●) sur le premier axe. (b) Sélection des méta-nœuds pour les plus fortes valeurs sur le 5e axe.

FIGURE 5.18 – Sélections de méta-nœuds dans l'affichage *distribution*.

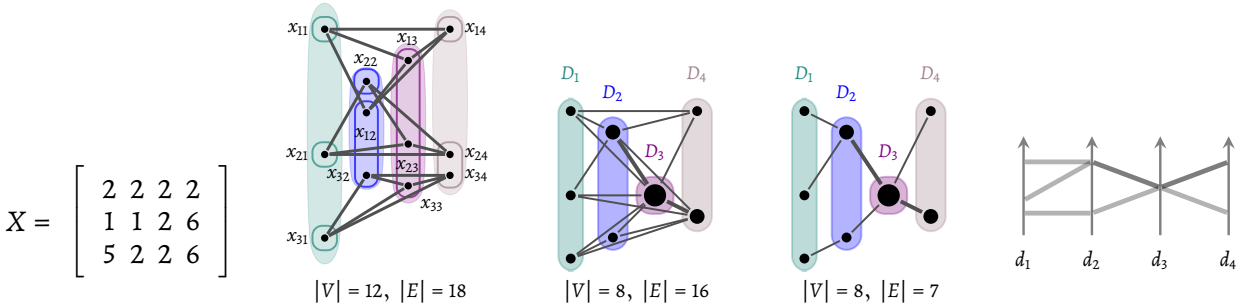
extrémités. Les méta-arêtes sont rattachées sur leurs extrémités à la position verticale correspondant au point de plus forte densité. Pour les deux encodages, les méta-nœuds contiennent un histogramme lissé détaillant la distribution des valeurs qu'ils couvrent ④ et l'espacement inter-méta-nœuds peut-être comparé au sein d'un même axe ④. Enfin, les espaces verticaux séparant les méta-nœuds sont proportionnels dans les deux encodages à la taille des intervalles de valeurs correspondant.

Les deux encodages sont présentés pour un jeu de données réel sur la figure 5.17. La transition d'un encodage à l'autre est animée et n'induit aucun croisement de méta-nœuds sur les axes puisque leur ordre reste le même. Cette interaction ne nécessite aucun échange avec le serveur. Les deux catégories d'interaction nécessitant l'apport de données du serveur sont le ré-ordonnement d'axes et la sélection d'entités. Les interactions de ré-ordonnement d'axes sont le déplacement, l'insertion et la suppression d'un ou plusieurs axes. Les interactions de sélections font apparaître des jauges sur chaque agrégat pour indiquer la proportion concernée par les entités sélectionnées. L'utilisateur peut définir une sélection d'entités en sélectionnant un méta-nœud (cf. figure 5.18a), une méta-arête ou une combinaison de méta-nœuds (cf. figure 5.18b). On appellera sélection *simple* les sélections de méta-nœud ou méta-arête et sélection *composées* celles combinant plusieurs méta-nœuds ou méta-arêtes.

5.3.2 *L'agrégation dans le formalisme graphe*

Dans cette section, nous étendons le formalisme de graphe présenté en section 5.2.2 pour l'adapter aux coordonnées parallèles abstraites. Le but de ce formalisme est de faire le lien entre les données originelles et la structure des représentations abstraites de coordonnées parallèles. Faire ce lien est utile pour formaliser les manipulations opérées par les interactions sur les données mais aussi pour énumérer le nombre d'états d'interaction possibles et ainsi envisager l'option du pré-calcul pour la réduction des latences de traitement.

Les représentations existantes de coordonnées parallèles abstraites s'appuient sur une agrégation des entités (mD) ou des valeurs de chaque dimension (1D) de la matrice d'entrée. Ces deux types d'agrégations correspondent à une agrégation des nœuds et des arêtes du formalisme graphe. La figure 5.19b présente un exemple d'une partition des nœuds de valeur identique sur



(a) Matrice X. (b) G_X avec R. (c) $G_{X/R}$. (d) Sous-graphe $G_{X/R}$. (e) Rendu par transparence.

FIGURE 5.19: Agrégation d'éléments aux valeurs identiques pour reconstruire par pré-agrégation le rendu par transparence. (b) Graphe pour X augmenté de la partition R groupant les valeurs identiques sur une même dimension. (c) Graphe quotient $G_{X/R}$. (d) Sous-graphe $G_{X/R}$ pour la séquence de dimension $(d_1,\ d_2,\ d_3,\ d_4)$. (e) Rendu par transparence obtenu en encodant le poids des arêtes du graphe quotient par l'opacité des lignes.

la même dimension pour la matrice X. Tous les nœuds de D_3 sont par exemple dans la même partie de R puisque la troisième colonne de X ne comporte qu'une seule valeur. Le processus d'abstraction du graphe selon une telle partition R consiste à fusionner les nœuds appartenant à la même partie dans R et à fusionner les arêtes en conséquence. Le graphe fusionné de l'exemple est dessiné sur la figure 5.19c. Ce processus correspond à construire le graphe quotient de G_X relatif à la partition R.

On note le graphe quotient de G_X, $G_{X/R} = (V,\ E, w_V, w_E, D)$ où :

— V est l'ensemble des méta-nœuds, correspondant aux parties de R,

— E l'ensemble des méta-arêtes,

— w_V est une valuation associant des propriétés aux méta-nœuds calculées à partir de la fonction w du graphe G_X,

— w_E est une valuation associant à chaque méta-arête le nombre d'arêtes qu'elle représente,

— D la partition de V groupant les méta-nœuds par dimension.

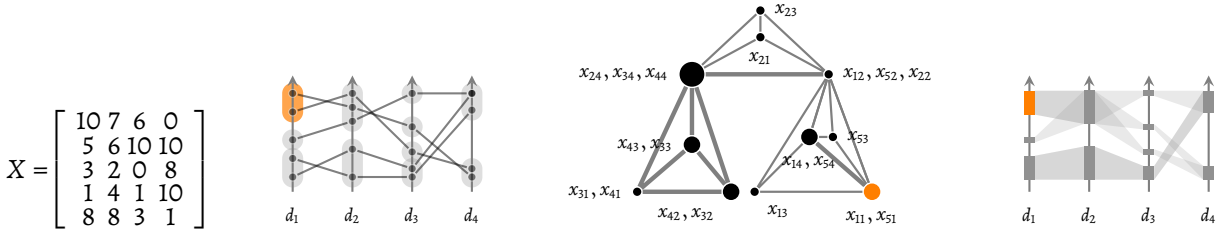
Pour conserver la propriété de m-partition qui permet de positionner les nœuds sur leur axe correspondant à partir de leur appartenance aux parties de D, R doit être un raffinement de D. De même que la fonction de valuation w est nécessaire pour placer verticalement les nœuds sur leur axe, la fonction w_V fournit les propriétés nécessaires à la représentation visuelle des méta-nœuds : les extrema, la moyenne ou encore la distribution des valeurs des nœuds sous-jacents, leur nombre, etc. La valuation w_E fournit les propriétés nécessaires à la représentation visuelle des méta-arêtes : leur poids uniquement d'après les encodages existants.

REPRÉSENTATION CONVENTIONNELLE PAR COMPOSITION DE LA TRANSPARENCE

La représentation de coordonnées parallèles au rendu par composition de la transparence peut être recomposée à partir du graphe quotient de la partition R groupant les éléments aux valeurs dimensionnelles égales comme illustré par les différentes étapes de la figure 5.19. L'agrégation permet de déterminer le nombre minimal de lignes à dessiner pour obtenir une vue identique à celle composée d'une polyligne par entités puisque l'agrégation dans l'espace des données est ici identique à l'abstraction à l'étape de rendu avec composition des couleurs des polylignes dans l'espace écran. Dans notre exemple, le sous-graphe $G_{X/R}[d_1, d_2, d_3, d_4]$ a deux méta-arêtes qui correspondent à la superposition de deux lignes sur le rendu par transparence.

REPRÉSENTATION ABSTRAITE PAR AGRÉGATION DES VALEURS DIMENSIONNELLES

Le processus est le même pour l'agrégation de valeurs dimensionnelles et est illustré sur la figure 5.20 qui présente la construction d'une vue abstraite utilisant les extrema chaque méta-nœud pour encoder leur hauteur et la cardinalité de chaque méta-arête pour encoder leur couleur.



(a) Matrice à 5 entités. (b) Partition R. (c) Graphe quotient $G_{X/R}$. (d) Représentation.

FIGURE 5.20: Des données à la représentation abstraite pour des partitionnements des valeurs par dimension (encodage *intervalle*).

REPRÉSENTATION ABSTRAITE PAR AGRÉGATION DES ENTITÉS Ce formalisme peut aussi être le support de coordonnées abstraites basées sur l'agrégation mD puisqu'un partitionnement des entités peut s'exprimer comme une partition R de V telle que $R \prec D$, en partitionnant chaque partie de D suivant le partitionnement des entités. La figure 5.21 présente un exemple de partitionnement des entités (en couleur) et de partition de V correspondante ainsi qu'une représentation similaire à celle de FUA et al. [62] qui peut être construite à partir du graphe quotient de R. Chaque groupe d'entités est représenté par un polygone délimitant les valeurs extrêmes du groupe sur chaque axe et une polyligne représentant l'entité moyenne du groupe.

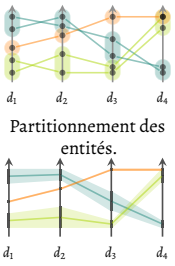


FIGURE 5.21: Agrégation des entités et représentation par rubans polygonaux.

5.3.3 Évaluation des interactions

Nous présentons ici une évaluation de la taille d'une abstraction (nombre de nœuds et nombre d'arêtes) et de ses interactions de déplacement d'axes et de la sélection. Cette taille influe à la fois sur le coût de stockage (et de calcul) lorsque ces interactions sont pré-calculées, et sur les temps de transfert.

Principe du pipeline conceptuel

Pour une matrice X et une partition R de son graphe G_X, un état de la visualisation est défini par l'historique des interactions, si l'on écarte les paramètres de rendu (niveau de zoom, gradient de couleur, etc). La figure 5.22 illustre le lien entre les interactions et la construction d'une vue en s'appuyant sur le formalisme graphe. Ce pipeline est conceptuel et ne schématise pas la structure d'une implémentation efficace mais plutôt les interconnections entre interactions.

Les interactions de sélections reviennent à la définition d'un sous-ensemble des entités d'intérêt qui correspond à un sous-ensemble S des nœuds de G_X. Lorsqu'une sélection est active, les propriétés du sous-graphe $G_{X/R}[S]$ sont utilisées pour déterminer les données support de la représentation de la sélection comme la contribution des entités sélectionnées dans le poids, les extrema de la sélection, etc. Les interactions de ré-ordonnement (déplacement, insertion, suppression) reviennent à l'édition de la séquence $a = (d_i, \ldots, d_j)$ d'axes représentée et modifient le filtrage des arêtes du graphe quotient finalement transféré au client.

Ce système d'interactions nous permet d'évaluer le nombre d'états d'interactions possibles.

Évaluation de la taille et du nombre d'états d'interaction

On suppose que le nombre de parties de la partition R subdivisant chaque partie de D est borné par un paramètre $k \in \mathbb{N}$, qui conditionne la granularité d'abstraction. Alors, on peut estimer la taille d'un état en nombre de nœuds et arête du graphe quotient correspondant. Le nombre maximal de nœuds d'un graphe quotient après filtre des arêtes pour m axes est :

$$N_{\text{nœuds}} \leqslant km, \tag{5.1}$$

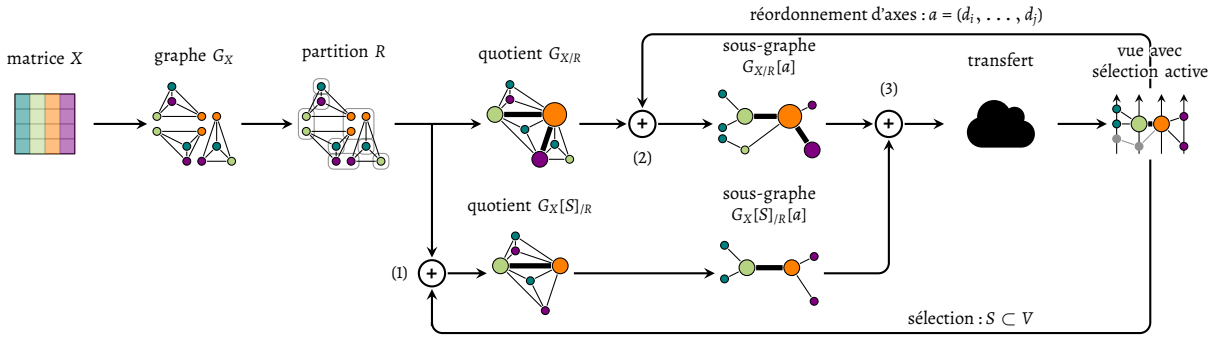


FIGURE 5.22: Pipeline illustrant le lien entre les interactions utilisateur et les étapes de construction du graphe codant la structure de la visualisation. Les interactions paramétrisent différentes étapes de la construction de la structure finale. La sélection (1) résulte en le calcul de propriétés des méta-nœuds et arêtes sur un sous-ensemble des entités, combiné ensuite aux données abstraites (3). Le déplacement d'axes condition le filtre des arêtes des graphes quotients (2).

et son nombre maximal d'arêtes est celui d'un graphe m-parti complet à m nœuds par partie, soit :

$$N_{\text{arêtes}} \leqslant \frac{k^2 m(m-1)}{2} = O(k^2 m^2). \tag{5.2}$$

Le stockage nécessaire pour supporter un système n'incluant que les ré-ordonnements d'axes est donc de $N_{\text{nœuds}} + N_{\text{arêtes}} = O(k^2 m^2)$. Chacun de ces nœuds et arêtes peut être à l'origine d'une sélection simple et chaque sélection simple implique potentiellement tous les nœuds et arêtes de ce graphes. Il y a donc au plus $N_{\text{nœuds}} + N_{\text{arêtes}}$ sélections simples possibles et leur pré-calcul a un coup de stockage borné par $O((N_{\text{arêtes}} + N_{\text{nœuds}})^2) = O(k^4 m^4)$ éléments ce qui est raisonnable pour une dizaine de dimensions et une valeur de k inférieure à la centaine.

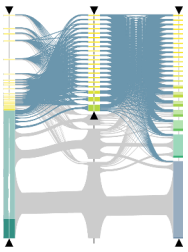


FIGURE 5.23: Sélection *complexe* par déplacement de paires de curseurs sur chaque axes.

Les sélections dites *complexes* fonctionnent par déplacement d'une paire de curseurs sur chacun des axes. La sélection est construite par l'intersection des ensembles d'entités correspondant aux nœuds encadrés par les curseurs sur chaque axe (cf. figure 5.23). Le nombre de sélections formées de p nœuds consécutifs sur un axe est le nombre de sous-chaînes de caractères de longueur p d'une chaîne de caractères de longueur k, soit $k - p$ donc le nombre total d'intervalles de 1 à k nœuds est :

$$\sum_{p=1}^{k} k - p = \frac{1}{2}k(k+1) = O(k^2). \tag{5.3}$$

Ainsi, le nombre de sélections *complexes* sélectionnant des nœuds par curseur pour chacun des m axes est donc $O(k^{2m})$ ce qui représente un coût prohibitif pour le pré-calcul et le stockage.

L'état initial requiert le transfert d'un graphe quotient $G_{X/R} = (V, E)$ de taille $|V| = O(km)$ et $|E| = O(k^2 m)$. Le déplacement, l'insertion ou la suppression d'un axe nécessite de transférer au plus $O(k^2)$ arêtes et $O(k)$ nœuds. Une sélection nécessite le transfert d'un graphe quotient de taille au maximum du même ordre de grandeur que l'état initial, soit $|V| = O(km)$ et $|E| = O(k^2 m)$. Tant que k est choisi raisonnablement pour la représentation ($k < 200$), ces tailles sont compatibles avec un transfert réseau en temps interactif.

5.3.4 *Implémentation et évaluation des performances*

Dans cette première implémentation, deux formes de calcul d'interaction entrent en jeu : le requêtage de résultats pré-calculés et le calcul à la volée. Le calcul à la volée concerne les sélections complexes et le pré-calcul les sélections simples. Ce choix est justifié par le nombre de cas exponentiel de sélections complexes possibles relativement à m. Il est implémenté en utilisant Elasticsearch [3], un système distribué déployé sur les machines de la plateforme aux

3. https://www.elastic.co/products/elasticsearch

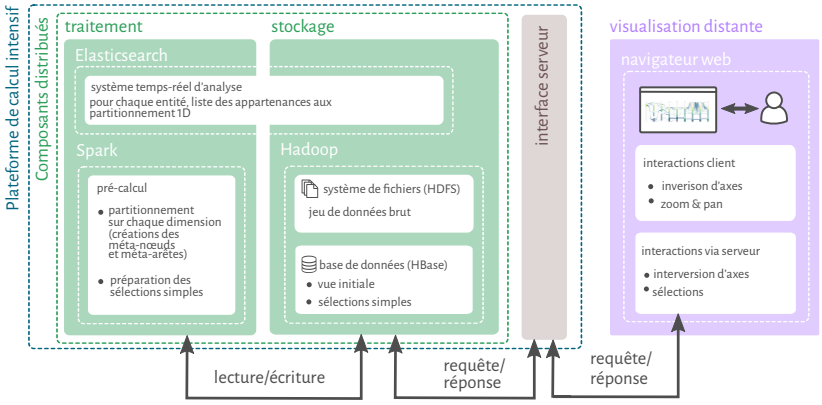


FIGURE 5.24 – Schéma du système de visualisation de coordonnées parallèles agrégées implémenté et évalué. La plateforme distante fait intervenir deux composants dans la réponse aux requêtes du client. Le composant Elasticsearch calcule les requêtes à la volée, et le composant HBase fournit les requêtes pré-calculées.

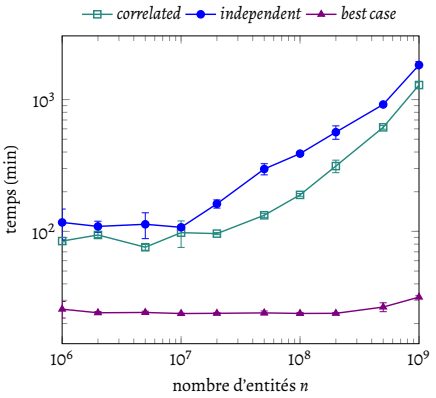
côtés de l'environnement Hadoop. La phase de pré-calcul concerne d'une part le partitionnement des données qui prépare le calcul à la volée et est préliminaire au pré-calcul de la vue initiale, incluant les arêtes de tous ses sous-espaces, ainsi que des sélections simples. Ces pré-calculs sont implémentés avec Spark et leurs résultats sont stockés dans HBase (cf. figure 5.24).

Périmètre d'évaluation

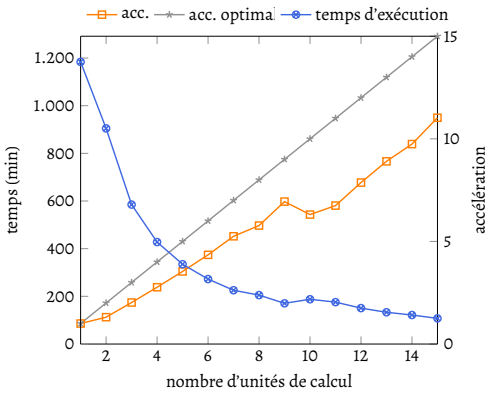
Dans un premier temps nous mesurons les performances de la phase préliminaire correspondant à la préparation d'un jeu de données. Dans un second temps, nous mesurons les latences d'interactions du système, c.-à-d. performances (temps d'exécution) des sélections simples et composées. Finalement nous examinons la scalabilité du système en mesurant l'accélération obtenue par l'allocation de ressources supplémentaires, dans les deux configurations de calcul. L'*accélération* (*speedup*) pour p unités de calcul correspond au ratio entre le temps d'exécution pour un nombre d'unités de calcul de référence p_{ref} (1 en général) et le temps d'exécution pour $p > p_{\text{ref}}$ unités de calcul.

Les résultats de ces expériences dépendent de plusieurs facteurs : en plus du nombre d'unités de calcul, des coûts de communication, le jeu de données et les paramètres d'abstraction (algorithme de partitionnement et k) entrent en compte. Nous considérons trois types de jeux de données synthétiques générés pour différents facteurs de corrélations entre paires de dimensions [22], et des tailles allant du million au milliard d'entités pour 15 dimensions et 15 agrégats par dimensions ($k = 15$). Le premier type de jeu de données est *independent* : chaque paire de dimensions a un coefficient de corrélation proche de zéro ce qui correspond à un grand nombre de méta-arêtes entre chaque paire de dimensions (proche de k^2). Ainsi ce type représente le pire des cas pour notre système. Le second type est *correlated*, généré de façon à obtenir un coefficient de corrélation d'au moins 0,60 pour chaque paire de dimensions. Ce type a pour but de s'apparenter à un jeu de données *réel*. Le dernier type, *best case*, possède le nombre minimal de méta-arêtes entre chaque paire de dimensions : k.

Pour chaque test, les mesures sont évaluées sur une grappe de 16 machines et moyennées sur trois exécutions. Les données sont transférées sur un réseau local entre le serveur et la plateforme ainsi qu'entre le client et le serveur. Les temps de réponses sont mesurés depuis le client, ainsi ils incluent le coût de transferts : de la plateforme au serveur et du serveur au client (cf. figure 5.24). Les requêtes de déplacement d'axes ne sont pas évaluées car elles fonctionnent de la même manière que les sélections simples tout en ayant des performances équivalentes ou meilleures.



(a) Temps d'exécution pour les trois jeux de données en fonction de n.



(b) Évaluation de la scalabilité avec *correlated*, $n = 10^7$.

FIGURE 5.25 – Performance de la phase préliminaire (partitionnement, calcul des propriétés des méta-nœuds et méta-arêtes, pré-calcul des sélections simples et insertion dans HBase).

Performance de la phase préliminaire

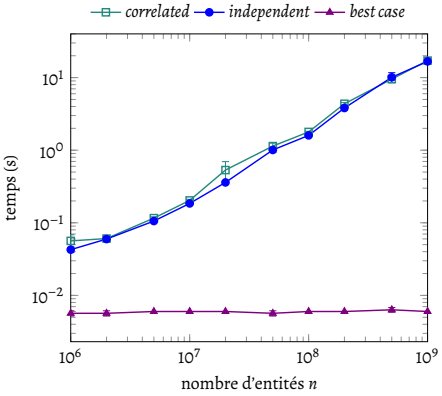
La phase préliminaire sert à la préparation d'un jeu de données et inclut le partitionnement des données, l'indexation des données annotées par le partitionnement dans Elasticsearch pour le calcul à la volée et le pré-calcul des propriétés des méta-nœuds (poids, extrema, distribution) et méta-arêtes (poids, extrémités) nécessaires au rendu de la vue initiale et des sélections simples, toutes séquences d'axes confondues. Les résultats de ces pré-calcul sont stockés dans HBase.

Ces temps de préparation, excluant l'étape d'indexation dans Elasticsearch, ont été mesurés pour différentes tailles de jeu de données. Les résultats sont présentés sur la figure 5.25a pour un n croissant et pour les trois types de jeux de données. Le jeu de données *best case* se distingue des deux autres au moins en partie car les k arêtes par paire d'axes réduisent considérablement le nombre et le coût du calcul des sélections simples puisque *correlated* et *independent* ont environ k^2 soit k fois plus ($k = 15$).

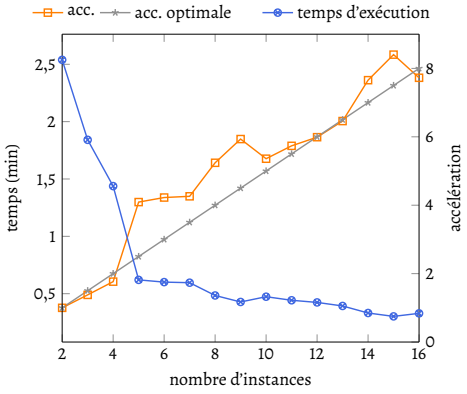
Nos observations montrent que les temps de traitement de la phase préliminaire sont dominés par l'étape de pré-calcul des sélections simples et que l'étape de partitionnement ne compte que pour 0,10 % du temps de traitement total en moyenne. Ces traitements sont coûteux : ils représentent plus de 32 heures pour les jeux de données d'un milliard d'entités de type *independent* et 24 heures pour *correlated*. Puisque cette étape s'effectue de manière distribuée, elle peut être accélérée en augmentant le nombre d'unités de calculs participant à cette tâche. Nous avons évalué la scalabilité de la phase préliminaire en mesurant son temps d'exécution pour un nombre d'unités de calcul croissant. La figure 5.25b présente le temps d'exécution moyen et l'accélération correspondante pour le jeu de données *correlated* avec 100 millions d'entités. L'accélération est mesurée relativement à l'utilisation d'une unique unité de calcul (exécution séquentielle). Au pire, celle-ci est juste au-dessus de 60 % de l'accélération optimale ce qui indique que le coût des communications est raisonnable, et son évolution démontre une bonne scalabilité malgré un palier autour de dix exécuteurs.

Performance des interactions par requêtes à la volée

Les requêtes de sélections composées sont traitées par calcul à la volée avec Elasticsearch et sont implémentées par un filtre suivi d'une agrégation. Ainsi, leur complexité dépend de la taille du sous-ensemble d'entités sélectionnées. Pour évaluer les performances des interactions de sélections composées, nous évaluons la requête qui est jugée la plus coûteuse : la sélection de tous les agrégats qui implique d'effectuer la ré-agrégation d'une vue initiale. Les résultats de cette évaluation sont présentés sur la figure 5.26a. Les temps de réponse augmentent linéairement avec le nombre d'entités comme attendu et reste en deçà d'une seconde pour les jeux de données



(a) Temps de réponse en fonction de n pour les trois jeux de données.



(b) Évaluation de la scalabilité avec *independent*, $n = 2 \cdot 10^8$.

FIGURE 5.26 – Évaluation des performance de la sélection complexe, dans son pire cas, calculée à la volée avec Elasticsearch. (a) Temps de réponse de la construction de la vue initiale. (b) Évaluation des performances des requêtes de sélection de méta-nœud pour le jeu de données *independent*.

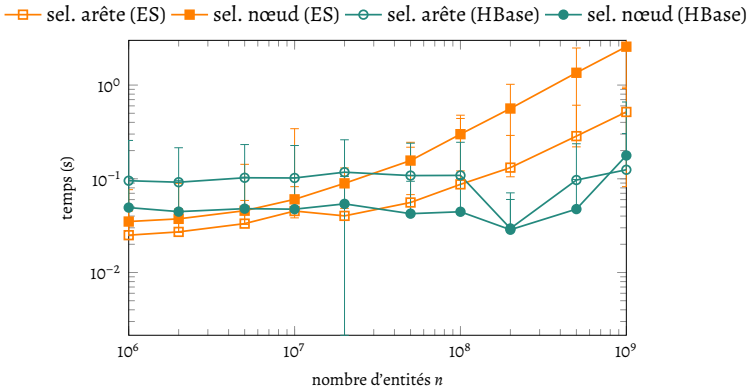


FIGURE 5.27 – Comparaison du temps de requête des données pré-calculées (HBase) et du temps de calcul à la volée (ES) pour les sélections simples possibles sur une vue initiale des jeux de données *correlated* en fonction de n.

de moins d'un milliard d'entités. Cependant, ils atteignent 15 secondes pour un milliard d'entités. Cela indique que les dimensions de la plateforme (16 machines) ne sont pas suffisantes (pour les jeux de données *correlated* et *independent*) et plus de ressources sont donc nécessaires pour assurer l'interactivité.

Les performances très basses de *best case* résultent de la pré-agrégation en amont des valeurs identiques qui ne sont insérées qu'une seule fois lors de l'indexation dans Elasticsearch. Par conséquent, l'étape d'agrégation est appliquée sur de petites données pour *best case*.

Notre plateforme montre ses limites pour les plus gros des jeux de données de test. Nous évaluons la scalabilité des requêtes sur Elasticsearch en mesurant l'accélération relativement à l'utilisation de deux unités de calcul. Les résultats présentés sur la figure 5.26b montrent les temps de réponse de la même requête, sur un jeu de données *independent* uniquement, avec 2 milliards d'entités. Ici l'accélération apparaît parfois meilleure que l'optimale calculée à partir de la version à deux unités de calcul.

Performance des interactions par requêtes pré-calculées

Les requêtes de sélections simples sont pré-calculées et leur résultat est requêté depuis HBase pendant l'interaction. Pour évaluer les performances de ces requêtes, nous avons effectué un test comparatif les comparant aux performances du calcul à la volée (Elasticsearch), pour les même

requêtes et le même nombre d'unités de calcul. L'objectif de cette expérience est de caractériser le gain apporté par le pré-calcul et en particulier les échelles pour lesquelles la tendance s'inverse. Pour différents jeux de données de différentes tailles, nous avons mesuré les temps de réponse des sélections simples possibles sur une vue initiale, c'est-à-dire une pour chaque agrégat présent sur cette vue. Ces résultats ont été moyennés séparément pour les sélections de méta-nœud et les sélections de méta-arête. La figure 5.27 compile les résultats pour les jeux de données *correlated* (*independent* présente des résultats similaires). De manière générale, les requêtes préparées permettent d'obtenir une réponse en moins de *0,10* seconde depuis HBase et ce temps ne semble pas s'accroître avec le nombre d'entités pour les valeurs testées. En revanche, comme attendu, les temps de réponse d'Elasticsearch augmentent jusqu'à excéder la seconde pour les jeux de données $n > 5 \cdot 10^8$. Cela montre qu'à partir d'une certaine taille de jeu de données, le gain du pré-calcul est important.

Les coordonnées parallèles abstraites permettent de fournir une vue d'ensemble d'un jeu de données, quelle que soit sa taille, puisque l'agrégation réduit le problème de la représentation à la visualisation de données de petite taille. De manière générale, l'agrégation réduit les temps de rendu et nous avons aussi montré qu'elle permettait de réduire les temps d'interaction pour certaines sélections qu'il devient envisageable de pré-calculer avec une infrastructure de traitement distribué. Malgré les interactions de sélections qui permettent de tracer des sous-ensembles d'entités, l'agrégation limite inéluctablement la granularité des analyses possibles. En effet l'utilisateur est cantonné à une abstraction unique, c'est-à-dire une unique manière d'agréger les données, et dépourvu de mécanisme d'obtention de détail supplémentaire (changement de niveau de détail). Dans la suite du chapitre, nous présentons une amélioration de représentation de coordonnées parallèles abstraites associé à des interactions de navigation multi-échelle permettant d'accéder au détail localement.

5.4 Coordonnées parallèles hiérarchiques

Dans cette section on s'intéresse aux représentations et aux interactions permettant à l'utilisateur de naviguer dans des représentations abstraites pour remédier aux limitations de l'abstraction : soit en changeant le niveau de détail (partition plus fine), soit en ajustant la partition dont résulte l'abstraction. Dans un premier temps, nous présentons les interactions de changement de niveau de détail présentées dans des travaux existants (section 5.4.1). Ensuite nous décrivons comment ces approches s'expriment dans le formalisme graphe présenté dans les deux sections précédentes (section 5.4.2) et nous comparons différentes interactions de navigation au regard des problématiques de scalabilité (section 5.4.3). Enfin, nous présentons en section 5.4.4 le second système implémenté et ses performances.

5.4.1 *Espace de conception d'interactions de navigation multi-échelle*

Le premier type d'interaction assurant le changement de granularité de l'agrégation, c.-à-d. de *niveau de détail*, consiste à calculer l'abstraction à une granularité croissante de sorte que l'utilisateur puisse changer de niveau de détail [129, 127, 124]. Dans le cas des partitions par *binning* [127], différents niveaux de détails peuvent être calculés en utilisant une grille de plus en plus fine (cf. figure 5.28a). Dans le cas de partitionnement basé sur des fonctions d'estimations de la densité (KDE) [129], différents niveaux de détails sont obtenus en changeant la force du lissage. Cette approche se décline également avec l'utilisation des niveaux d'une hiérarchie pour définir des niveaux de granularité différents [25] (cf. figure 5.28b). Dans ce contexte, on emploie les termes de *drill-down* (respectivement *roll-up*) pour désigner une interaction passant au niveau supérieur (respectivement inférieur) dans la hiérarchie, c.-à-d. à un niveau de détail plus élevé (respectivement moins élevé). Le désavantage de cette approche est qu'à mesure que l'utilisateur augmente le niveau de détail le nombre d'éléments graphiques augmente également. Ainsi, pour atteindre un niveau de détail permettant de distinguer les entités par exemple, il est nécessaire

de passer par une représentation dont le nombre d'éléments graphiques avoisine celui de la représentation conventionnelle ce qui élimine les avantages de l'agrégation. Dans le cas des travaux de PALMAS et al. [129], le changement de niveau de détail est limité à un axe plutôt qu'à toute la représentation mais possède la même limitation d'encombrement visuel pour la visualisation de grand nombre d'entités.

Une manière d'éviter l'explosion du nombre d'éléments graphiques est d'associer un filtre au changement de niveau de détail. Cette approche s'apparente au zoom géométrique des cartes géographiques interactives qui augmente le niveau de détail tout en rognant les bords d'une image pour obéir aux contraintes de taille de la vue. Cette approche est celle de l'interaction de *drill-down* de VOSOUGH et al. [161] où l'ensemble des entités sont filtrées à chaque sélection d'un nœud d'une hiérarchie dimensionnelle (cf. figure 5.28c). Le désavantage de cette approche est la perte du contexte. Le filtrage croisé qui s'opère à la sélection consécutive de nœuds (d'un même axe ou de plusieurs axes) réduit au fur et à mesure la quantité d'entités représentées et ne préserve comme contexte qu'une version réduite de la hiérarchie sans les liens filtrés.

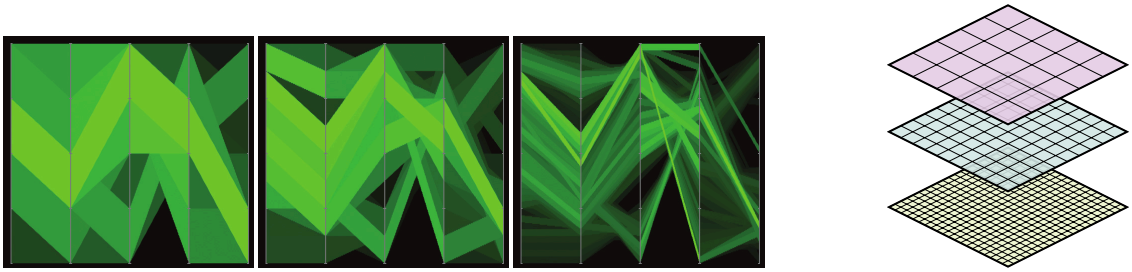
Une autre manière de contrôler le nombre d'éléments graphiques est d'utiliser des coupures non équilibrées de la hiérarchie comme partitions. FUA et al. [62] utilisent le poids de chaque nœud de la hiérarchie, c.-à-d. le nombre de feuilles qu'il couvre, pour définir des coupures progressivement plus fine. La coupure à un certain niveau de détail μ est définie comme l'ensemble des nœuds dont le poids est inférieur ou égal à μ s'il n'est pas une feuille et dont le parent a un poids supérieur à μ. Cette stratégie est illustrée par trois coupures sur la figure 5.28d. Elle autorise plus de niveaux de détail différents que les niveaux de la hiérarchie et peut contribuer à réduire la quantité de changement lors d'un drill-down ou roll-up. Cependant elle ne permet pas de limiter le nombre d'éléments graphiques des vues notamment celui des vues à haut niveau de détail. Dans la suite de cette section, nous nous intéresserons aussi à d'autres types d'interactions de drill-down basées sur des coupures non équilibrées qui cherchent à limiter le nombre d'éléments graphiques de ces vues à haut niveau de détail.

5.4.2 *Changement de niveau de détail dans le formalisme graphe*

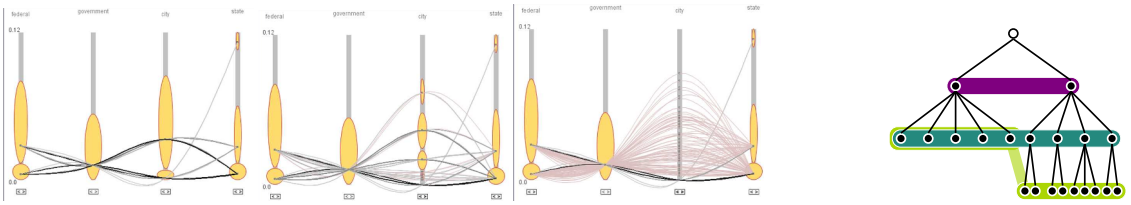
Pour permettre une navigation multi-échelle dans les coordonnées abstraites, nous étendons le formalisme graphe pour qu'il inclue une hiérarchie, arbre de partition du graphe. Les coupures de cette hiérarchie définissent des partitions des nœuds du graphe et des coupures partageant qui se raffinent l'une l'autre sans être entièrement différente ce qui correspond à un changement visuel « modéré » entre une abstraction et l'autre. Pour que les coupures de cette hiérarchie soient des raffinements de D, cette hiérarchie doit être l'union de m arbres $T_1, \ldots, T_m$, tels que T_i a pour feuilles les entités de D_i. Pour résumer, une matrice X est modélisée par un graphe $G_X = (V, E, w, D, T)$ où T est une hiérarchie issue de l'union de ces arbres et dont les feuilles sont les nœuds de V. Une abstraction de ce graphe selon cette hiérarchie est définie par une coupure de T. Ainsi la hiérarchie T permet de générer en théorie de multiple abstractions et on s'intéresse donc aux interactions de navigation entre différentes abstractions.

5.4.3 *Évaluation des interactions*

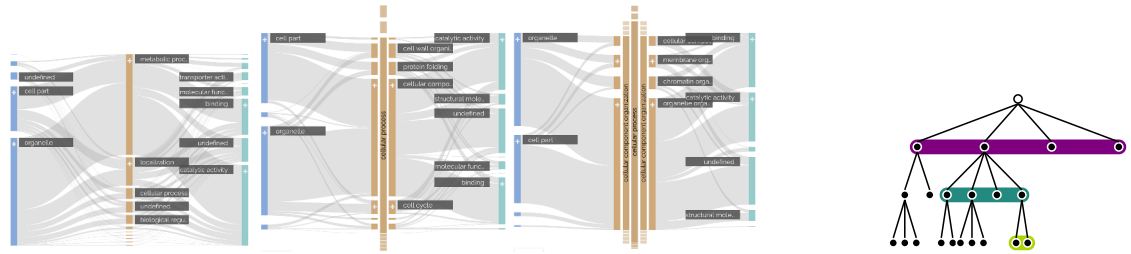
Pour un graphe $G_X = (V, E, w, D, T)$, une vue abstraite est définie par une coupure de T, une séquence de dimensions et, optionnellement, un sous-ensemble d'entités à mettre en surbrillance. La figure 5.29 montre comment ces paramètres interviennent dans la construction de la structure d'une vue abstraite. Les interactions de navigation conditionnent la partition R de V utilisée pour calculer le graphe quotient (1). Si une sélection est active, le graphe quotient de son sous-graphe induit est aussi calculé pour être présenté superposé à la vue (2). Ensuite, ces graphes quotients sont filtrés (3) pour ne conserver que les arêtes pertinentes relativement à la séquence de dimensions affichée et modifiée par les déplacements d'axes par exemple. Enfin, cette structure



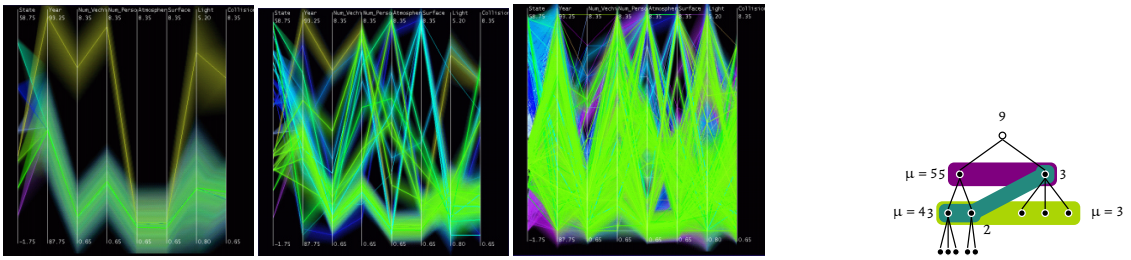
(a) Changement de niveau de détail (sans hiérarchie) [127].



(b) Changement de niveau de détail dirigé par une hiérarchie (une hiérarchie par axe) [25].



(c) Drill-down+filtre selon une hiérarchie (une hiérarchie par axe) [161].



(d) Coupure non équilibrée (selon une hiérarchie sur les entités) [62].

FIGURE 5.28: Trois niveaux de détail pour différentes approches de la navigation multi-échelle.

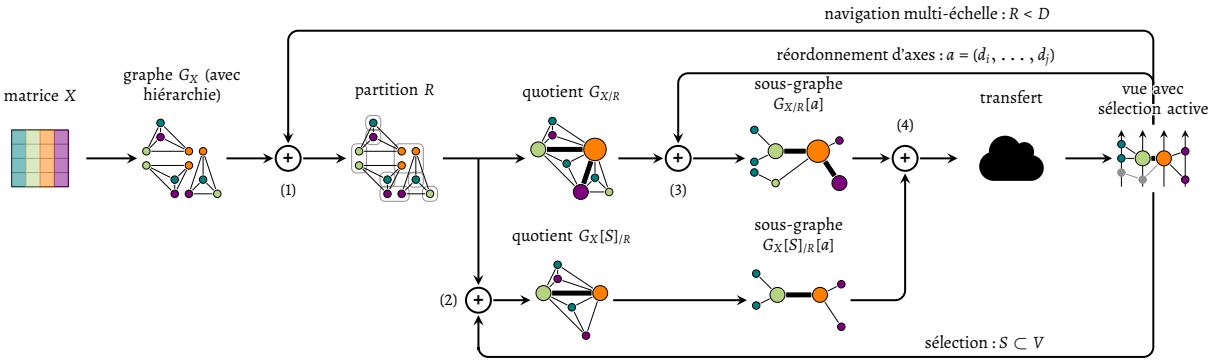


FIGURE 5.29: Pipeline illustrant le lien entre les interactions utilisateur et les étapes de construction du graphe codant la structure de la visualisation. Les interactions paramétrisent différentes étapes de la construction du graphe de la vue : la navigation hiérarchique opère en (1) dans la sélection de la partition R, la sélection opère en (2), et enfin le ré-ordonnement en (3).

agrégée et filtrée correspond à ce qui est transféré au client dans un contexte de visualisation à distance.

La limitation principale des coordonnées parallèles abstraites est la quantité restreinte d'information qu'une abstraction communique. L'utilisation d'une hiérarchie est une manière usuelle d'implémenter des transitions entre différents niveaux de détail. Une hiérarchie fournit aussi la structure nécessaire à des changements *locaux* de détail résultant en des vues combinant des entités de différentes granularité. Les opérations de *drill-down* (aussi appelées *ouverture*) et *roll-up* (aussi appelées *fermeture*) sont des opérations complémentaires de changement du niveau de détail. L'opération de drill-down consiste à inclure des nœuds plus profond, c.-à-d. plus détaillés, alors que l'opération de roll-up remplace des nœuds par d'autres moins profonds, c.-à-d. plus grossiers.

Dans notre contexte, on s'intéresse également aux interactions qui peuvent être calculées ou pré-calculées efficacement et qui permettent de maintenir un nombre raisonnable d'éléments graphiques dans la représentation. Notamment, il est important que les changements incrémentaux induits par une interaction soit de taille $\ll n$ pour être transférés. Concernant le pré-calcul des états de la navigation, en général le nombre de coupures d'un arbre est exponentiel relativement au nombre de feuilles. Par exemple, les arbres binaires complets, c.-à-d. dont tous les niveaux sont pleins sauf le dernier, ont $\Omega(2^n)$ coupures où n est le nombre de feuilles. Ceci rend le pré-calcul de tous les graphes quotients définis par l'ensemble des coupures issues d'un arbre impossible. Concernant le calcul des interactions, il est intéressant de noter que si les fonctions d'agrégation calculant les valuations des méta-nœuds sont associatives, alors une partie des opérations de roll-up peut être calculée à partir des nœuds de l'état précédent plutôt qu'à partir des données complètes.

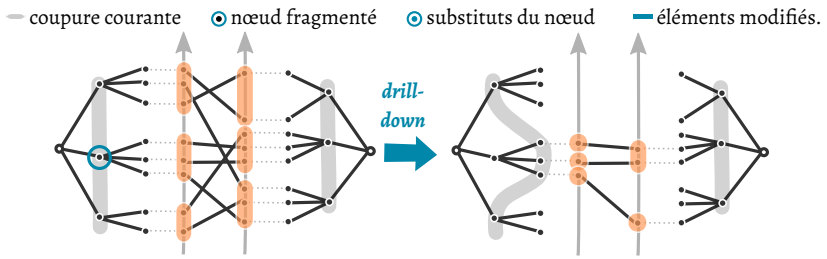
Comme vu précédemment, une opération qui augmenterait le niveau de détail globalement, en remplaçant chaque nœud de la coupure par ses enfants, aboutirait à une augmentation exponentielle du nombre d'entités visuelles, indépendamment de l'arité et de la profondeur de la hiérarchie. De plus, les changements incrémentaux correspondant aux données à transférer au client seraient exponentiellement plus grands à mesure que la coupure s'approcherait des feuilles. Ainsi cette approche ne satisfait pas les contraintes de notre contexte. Ouvrir un seul nœud à la fois réduit la taille des changements incrémentaux mais laisse à l'utilisateur la responsabilité de réguler le nombre de nœuds ouverts à l'écran en refermant les nœuds qui ne sont plus intéressants.

Dans ce qui suit, nous détaillons trois stratégies de navigation hiérarchique qui visent à limiter l'augmentation du nombre d'entités visuelles à l'écran à mesure que l'utilisateur augmente le niveau de détail.La figure 5.30 illustre ces approches sur un cas à deux dimensions. Chaque entité est représentée par une arête connectant une paire de nœuds, un sur chaque axe. Le premier axe (respectivement second axe) est augmenté par sa sous-hiérarchie T_1 (respectivement T_2) dont les feuilles sont les nœuds de D_1 (respectivement D_2). Sur ces sous-hiérarchies est représentée la coupure *courante* (en gris), c.-à-d. celle dont est issue la vue avant interaction. La partition des nœuds correspondant à cette coupure est montrée par des régions orangées.

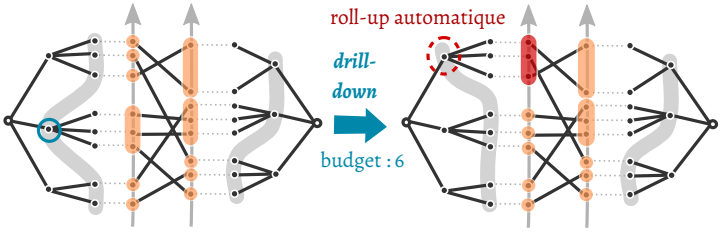
Détail et filtre

L'approche *détail et filtre* consiste à réguler le nombre d'entités visuelles à l'écran en combinant à l'opération d'ouverture d'un nœud un filtre éliminant les autres nœuds de la même dimension et en même temps les entités couvertes par ces nœuds (cf. figure 5.30a). Cette opération correspond à calculer le graphe quotient du sous-graphe induit par les entités couvertes par le nœud ouvert uniquement. Simultanément au filtre, le nœud ouvert est remplacé par ses enfants dans la coupure. Dans l'exemple de la figure 5.30a, le nœud sélectionné couvre trois entités et possède trois enfants dans la hiérarchie. Après l'opération, seules ces entités participent à l'abstraction et elles sont agrégées en trois méta-nœuds sur le premier axe et deux méta-nœuds sur le second.

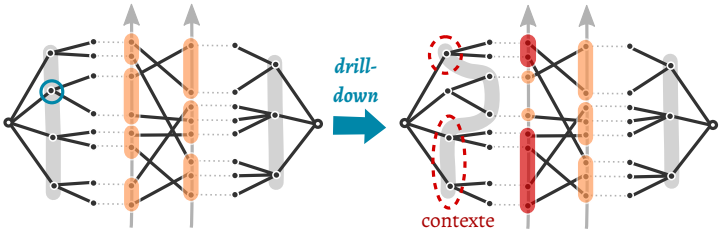
L'approche détail et filtre borne le nombre de méta-nœuds par axe par l'arité de la hiérarchie. En effet, aux cours des interactions, la représentation ne représente pas plus de nœuds par axe que le minimum entre l'arité de la hiérarchie et le nombre de nœuds par axe de l'état initial.



(a) Détail & filtre. Les nœuds correspondant aux entités non-couvertes par le nœud ouvert sont écartés : la vue résultant correspond à trois entités.



(b) Détail sous budget. Les trois premiers nœuds ont été remplacés par leur parent (en rouge à droite) pour respecter le budget après l'ajout des trois nœuds enfants du nœud sélectionné (entouré en bleu à gauche).



(c) Agrégation du contexte. Les nœuds de la coupure qui sont les plus profonds sont le focus (en orange). Les autres nœuds de la coupure sont repliés en deux nœuds de contexte (en rouge).

FIGURE 5.30 – Trois stratégies de navigation multi-échelle présentée pour deux dimensions.

Par conséquent, cela borne également le nombre de méta-arêtes ce qui offre une garantie sur la taille des changements incrémentaux à transférer pour cette opération, mais également pour les interactions de sélection et ré-ordonnement. Comme remarqué dans la section précédente, cette interaction élimine successivement les entités non sélectionnées de l'ensemble de la représentation à la fois ce qui supprime une part importante de l'information contextuelle de navigation. De plus, ce filtre nécessite la mise à jour des propriétés visuelles de tous les méta-nœuds et méta-arêtes restant sur la vue, comme les poids et intervalles des deux méta-nœuds du second axe sur notre exemple. Cette interaction est donc susceptible d'induire des changements visuels drastiques et donc difficiles à appréhender sur tous les sous-espaces de la représentation.

Détail sous budget

L'approche de détail sous budget consiste à utiliser des coupures non équilibrées dont la taille en nombre de nœuds est maintenue sous un certain budget prédéfini (cf. figure 5.30b). En effet, la taille de la coupure courante est directement liée au nombre de méta-nœuds du graphe quotient, lui-même conditionnant le nombre maximal de méta-arêtes et ainsi à la taille de l'abstraction transférée. Une demande de détail sur un nœud v consiste à remplacer ce nœud de la coupure par ses enfants ce qui augmente la taille de la coupure courante de $\mathit{enfants}(v) - 1$. Cet accroissement de la taille de la coupure implique potentiellement de modifier le reste de la coupure en de multiples points pour accommoder le budget prédéfini si celui-ci est alors dépassé. Par exemple, sur la

figure 5.30b le budget a été fixé à 6 par axe (12 pour la coupure au total). Avant drill-down le premier et second axes montrent respectivement 6 et 5 nœuds. Arès drill-down sur le nœud sélectionné, le premier axe montrerait 8 nœuds. Pour respecter le budget, les trois nœuds supérieurs sont remplacés par leur parent ce qui résulte en 6 nœuds sur la coupure doublement modifiée du premier axe de l'illustration de droite. Le second axe reste inchangé. Si les modifications de coupure sont contraintes à une seule sous-hiérarchie par interaction (celle subissant l'interaction) alors cette implémentation du drill-down induit des changements visuels restreints à un ou deux espaces inter-axes de la représentation ce qui répond à une des limitations de l'interaction détail et filtre. Calculer les méta-nœuds et méta-arêtes de ces espaces inter-axes requiert une passe sur les entités en plus du coût de recherche d'une coupure satisfaisante.

Pour atteindre un niveau de détail assez fin pour révéler des entités, la coupure courante doit contenir une ou plusieurs feuilles. Le budget minimal qui permettrait de définir une telle coupure dépend de l'arité et de la profondeur de la hiérarchie. Pour une hiérarchie d'arité k, n feuilles et profondeur h, une coupure minimale en taille contenant au moins une feuille peut avoir entre h et kh nœuds. Pour les arbres binaires dont les nœuds possèdent uniquement 0 ou 2 enfants par exemple, la hauteur h est comprise entre $\lceil log_2(n) \rceil + 1$ et n. Pour permettre de naviguer jusqu'à révéler des entités, le budget doit donc être choisi relativement à la profondeur et l'arité de la hiérarchie. Par exemple, sur la figure 5.30b, le budget doit être au moins de 5 par axe pour permettre les coupures traversant chaque fratrie de feuilles pour les deux dimensions.

La scalabilité de cette interaction dépend de l'allure de la hiérarchie : lorsque la profondeur de la hiérarchie a le même ordre de grandeur que n, cette approche ne passe pas à l'échelle.

Agrégation du contexte

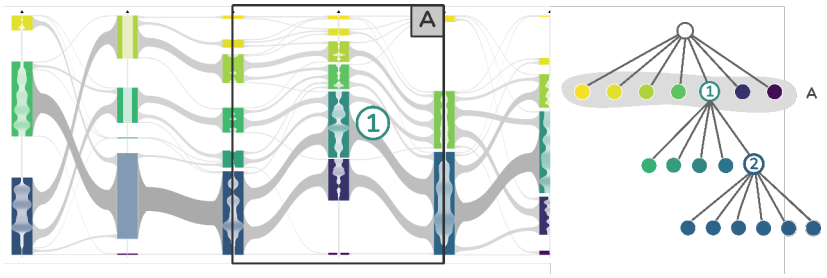
L'agrégation du contexte consiste à fusionner les méta-nœuds issus de nœuds hors du champ de focus, c.-à-d. non descendant du nœud sélectionné lors de l'interaction (cf. figure 5.30c). Cette agrégation réduit le nombre de méta-nœuds et amenuise l'impact des nœuds de *contexte*, c.-à-d. hors du champ de focus, sur le nombre d'éléments visuels. En général, ces agrégats ne correspondent pas directement à un nœud dans la hiérarchie, c.-à-d. ils ne possèdent pas d'ancêtre commun dans la hiérarchie. Par exemple, sur la figure 5.30, le nœud de contexte inférieur ne correspond pas à un unique nœud mais à deux. Cette approche permet de naviguer jusqu'au niveau des feuilles de la hiérarchie en limitant le nombre d'éléments visuels. En effet, avec f foci par axe et une hiérarchie d'arité k, chaque axe présente au plus $f - 1$ nœuds de contexte et fk nœuds de *focus*. Cette approche est celle implémentée, avec $f = 1$, dans le système décrit ensuite.

5.4.4 *Implémentation et évaluation des performances*

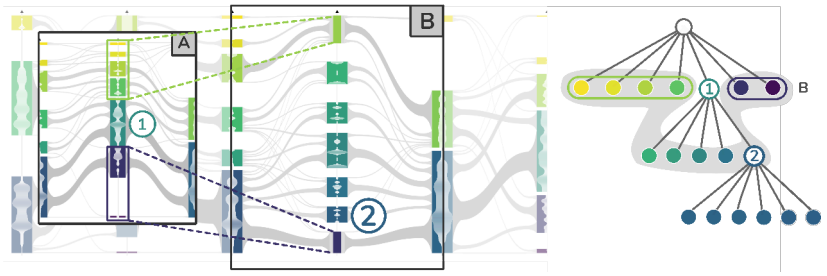
Les implémentations présentées dans cette section permettent la navigation multi-échelle en utilisant l'approche d'agrégation du contexte ainsi que les interactions de sélection et de déplacement d'axes par l'utilisation de calcul à la volée uniquement.

Vue focus+contexte

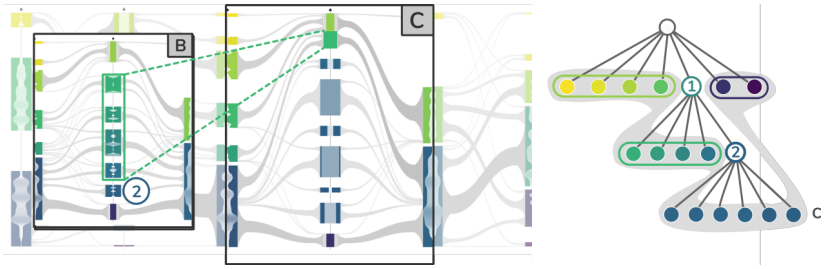
Pour les deux encodages présentés en section 5.3.1, nous utilisons la largeur des nœuds pour distinguer visuellement les nœuds de *contexte* des nœuds de *focus*. Similairement à un effet *fisheye*, les nœuds de focus sont présentés plus large donc plus proéminents que les nœuds de contexte (cf. figure 5.31b). Les intervalles couverts par les nœuds de focus, étant raffinées par chaque interaction de *drill-down*, les nœuds de focus tendent à couvrir à la fois des intervalles de plus en plus petits et correspondant à de moins en moins d'entités. Ainsi, pour garantir leur visibilité, ils sont rendus à une échelle différente des nœuds de contexte. La figure 5.31 présente un exemple de deux ouvertures successives sur le même axe sur la représentation *distribution*. Sur la figure 5.31b les nœuds de focus occupent approximativement 80 % de l'espace vertical alors que leur nœud parent, annoté ①, occupe environ 25 % de l'espace vertical sur la figure 5.31a. L'effet



(a) Vue initiale. Les nœuds de l'axe encadré sont en focus.



(b) Vue après l'ouverture du nœud ①. L'axe encadré a 5 nœuds de focus et deux nœuds contexte, chacun avec un bloc de niveau.



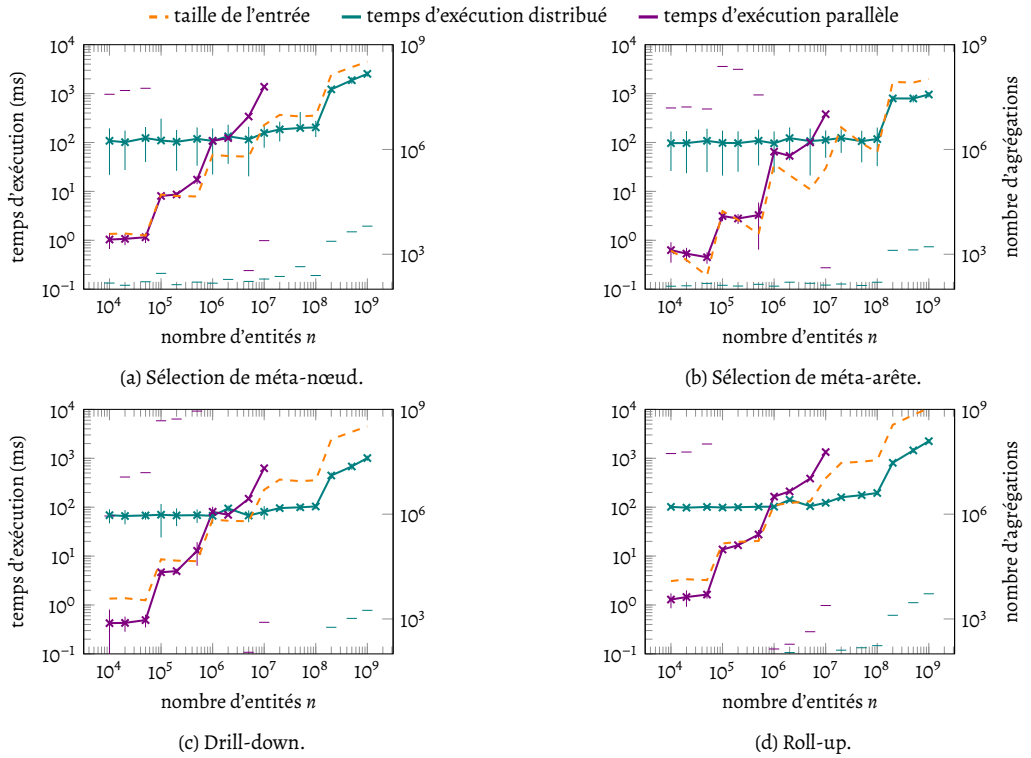
(c) Vue après ouverture de ②. L'axe encadré comporte six nœuds de focus et deux nœuds de contexte. Le nœud de contexte supérieur a deux blocs de niveaux et le nœud de contexte inférieur en a un.

FIGURE 5.31 – Vue focus+contexte. Ouvertures successives sur l'axe. L'arbre sur la droite est la hiérarchie de l'axe encadré et montre la coupure en gris. Cette illustration ne fait pas partie de la représentation et n'est montrée ici que pour faciliter la compréhension de l'interaction.

de grossissement se retrouve donc verticalement et horizontalement et la déformation verticale contribue à la scalabilité visuelle de l'encodage.

Les nœuds de contexte sont augmentés d'une pile de *blocs de niveau* qui résument le nombre d'opérations d'ouverture passées. Plus large ces blocs sont, plus profonds dans la hiérarchie sont les nœuds qu'ils représentent. Sur la figure 5.31c par exemple, le bloc intérieur supérieur est le plus large puisqu'il correspond à aux frères de ②. La hauteur et l'ordre vertical des blocs de niveau suit le même encodage que celui des nœuds de focus. Ces blocs sont calculés du côté client et ne sont donc pas transférés du serveur au client, ni comptés dans le budget en entités visuelles.

Dans le mode de changement de niveau de détail, l'utilisateur peut interagir avec les nœuds de focus (les ouvrir) et les blocs de niveaux (pour refermer les nœuds ouverts). Cliquer sur un nœud focus déclenche une animation durant laquelle l'espace vertical est déformé pour étendre le nœud, puis il est fragmenté en ses enfants et enfin ses nœuds frères sont compressées dans les nœuds de contexte. Cliquer sur un bloc permettent d'accéder à l'état dans lequel les nœuds qu'il représente sont en focus. Par exemple, sur la figure 5.31, cliquer sur le bloc violet sur les états C ou B amène l'utilisateur à l'état A puisque ce bloc représente deux nœuds frères de ①.



(a) Sélection de méta-nœud.

(b) Sélection de méta-arête.

(c) Drill-down.

(d) Roll-up.

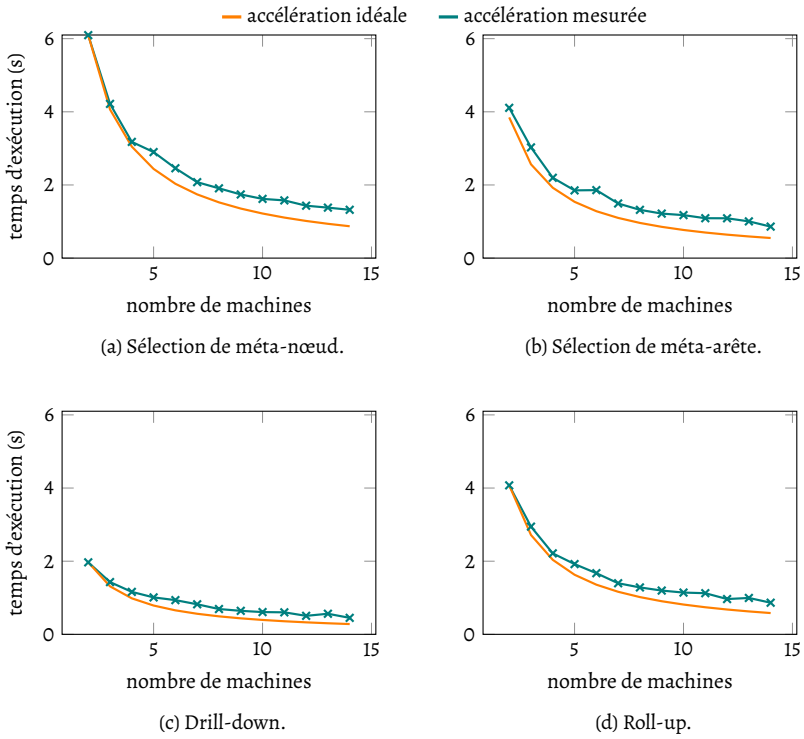
FIGURE 5.32 – Temps d'exécution pour les quatre opérations implémentées en fonction de n. Les mesures sont moyennées sur *1000* essais pour les deux implémentations (version distribuée et version parallèle).

Performance des traitements

À la différence de l'implémentation décrite dans la précédente section, dans la version multi-échelle du système toutes les interactions sont calculées à la volée. Nous effectuons des tests sur deux implémentations : une implémentation parallèle pour une unique machine et une implémentation distribuée pour une grappe de machine. L'implémentation distribuée est une application Spark exécutée en continu sur la plate-forme avec 15 machines. L'implémentation parallèle est une application C++/OpenMP exécutée sur une machine portable puissante avec quatre cœurs HT à *2,70 GHz* et *32 Go* de mémoire vive. Les jeux de données de test ont été générés pour différentes tailles entre 10^4 et 10^9 et pour 15 dimensions et $k = 32$. Les jeux de données utilisés sont de type *independent* ce qui tend à produire une nombre maximal de méta-arêtes. Les hiérarchies sont calculées en utilisant le partitionnement par canopées [113] appliqué de manière ascendante. Pour toutes les expériences, les mesures sont moyennées sur *1000* exécutions. Pour chaque type d'opération, nous évaluons l'interaction la plus coûteuse, c'est-à-dire celle agrégeant le plus d'entités.

Nous évaluons les temps d'exécution des opérations implémentées relativement à n. Le but de ces expérimentations est de confirmer que nous observons une augmentation linéaire dans les temps d'exécution à mesure que la charge augmente. Pour démontrer la scalabilité de l'implémentation distribuée système, nous évaluons aussi les performances relativement au nombre de ressources allouées.

TEMPS D'EXÉCUTION (DISTRIBUÉ ET PARALLÈLE) Nous comparons les temps d'exécution de l'implémentation distribuée et de l'implémentation parallèle pour quatre itérations : sélection de méta-arête, de méta-nœud, drill-down et roll-up et des tailles de jeu de données croissantes. La figure 5.32 montre le temps d'exécution moyen pour ces quatre interactions. Pour ces expérimentations, pour n inférieur à 10^6 approximativement, l'implémentation distribuée est moins performante que l'implémentation parallèle bien que cette dernière possède moins de



(a) Sélection de méta-nœud.

(b) Sélection de méta-arête.

(c) Drill-down.

(d) Roll-up.

FIGURE 5.33 – Test de scalabilité pour l'implémentation distribuée $n = 2 \cdot 10^8$ et entre 2 et 14 unités de calcul, chacune ayant 12 cœurs et *31 Go* de mémoire vive.

ressources. Jusqu'à $n = 10^8$, les quatre opérations s'exécutent en moins d'une seconde pour la version distribuée. Jusqu'à $n = 2 \cdot 10^8$, les performances de la version distribuée sont constantes en moyenne malgré l'augmentation de la charge. Jusqu'à cette limite, les temps d'exécutions sont aussi très variables, surtout pour les opérations de sélection. Le nombre d'entités agrégées pour chaque opération est représenté avec les courbes de temps d'exécutions (valeurs sur l'axe de droite). Cette courbe a une allure en escalier qui suggère que l'implémentation parallèle dépend linéairement de cette variable. On observe la même tendance pour l'implémentation distribuée à partir de $n = 10^8$. Cette allure en escalier peut s'expliquer par le fait que les hiérarchies possèdent un nombre variable de nœuds sur leur premier niveau et de manière générale le poids de leur nœud au poids le plus élevé n'augmente pas linéairement avec n. L'allure constante des performances de l'implémentation distribuée pour les plus petits des jeux de données n'est pas surprenante et suggère que les temps d'exécution pour ces jeux de données sont dominés par les coûts fixes liés aux entrées/sorties disque, aux échanges entre les machines de la grappe et au coût de l'agrégation finale des résultats partiels. Le coût d'agrégation des résultats reste approximativement constant puisqu'il est fonction de la taille du résultat et du nombre de tâches exécutées en parallèle.

Ces résultats montrent que, pour le dimensionnement des deux environnements d'exécution, l'implémentation parallèle est plus efficace pour les jeux de données de taille inférieure à 10^6. Au-delà, la plateforme distribuée est plus performante en moyenne et bénéficie d'autres atouts comme la résilience.

SCALABILITÉ HORIZONTALE DE LA VERSION DISTRIBUÉE On s'intéresse ensuite à l'évaluation de la scalabilité du système distribué en utilisant un jeu de données de plus de 2 milliards d'entités, ce qui correspond à une taille pour laquelle le temps d'exécution n'apparaît pas constant. Un des objectifs du système est de restreindre les traitements s'appliquant pendant l'interaction à des traitements à la complexité linéaire pour bénéficier pleinement de la scalabilité horizontale de l'infrastructure. Ainsi, on s'attend à ce que les résultats expérimentaux montre une accélération linéaire c.-à-d. que le temps d'exécution soit divisé par deux lorsque le nombre

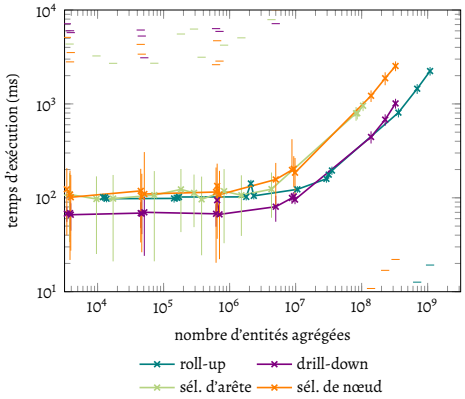


FIGURE 5.34 – Comparaison du temps d'exécution moyen entre les opérations en fonction du nombre d'entités (version distribuée).

d'exécuteurs est doublé. La figure 5.33 montre le temps d'exécution moyen pour quatre opérations calculées avec l'implémentation distribuée et un nombre croissant d'exécuteurs de 2 à 14. La courbe *idéale* pour le temps d'exécution correspond à une accélération linéaire relativement au temps d'exécution à deux exécuteurs. Ces résultats montrent que les performances sont proches de cet idéal.

COMPARAISON ENTRE OPÉRATIONS Pour finir, nous comparons les temps d'exécutions des opérations relativement à n. La figure 5.34 montre les performances des quatre opérations pour l'implémentation distribuée. Au-dessus de 100 millions d'entités traitées, les mesures forment deux groupes : d'un côté les opérations de sélection et de l'autre les opérations de navigation hiérarchiques (ouverture et repliement). Bien que ces courbes aient la même allure pour les deux groupes, les opérations de sélection apparaissent distinctement plus coûteuses par un facteur qui s'explique par le nombre de sous-espaces concernés par l'interaction. Les opérations de navigation hiérarchiques, telles qu'implémentées ne modifient qu'une dimension et donc n'affectent que deux sous-espaces au plus. Au contraire, les opérations de sélection affectent tous les sous-espaces représentés et sont donc plus coûteuses par un facteur en relation avec le nombre de dimensions à l'écran (15 axes dans nos expériences, soit 14 sous-espaces). Si nous implémentions les opérations d'éditions des hiérarchies, nous nous attendrions à ce que les opérations de fusion et fragmentations aient des performances comparables à celles des opérations de navigation hiérarchique.

5.5 Cas d'étude : enquête de recensement de 1990 aux États-Unis

Ce cas d'étude utilise un jeu de données public composé d'un échantillon de l'enquête de recensement de 1990 aux États-Unis, mis à disposition sur le répertoire de l'UCI Machine Learning [48]. Ce jeu de données est un échantillonnage à 1 % des archives du *Public Use Microdata Samples* relatives à ce recensement et comporte 125 dimensions pour environ 2,50 millions d'entités (individus). La plupart de ces dimensions sont de nature binaire (p. ex. *Worked in 1989*, *Language Other Than English*), d'autres sont catégoriels (p. ex. *Place of Birth*), ordinaux (p. ex. *Ability to Speak English*) ou numériques (p. ex. *Age*). Plusieurs hiérarchies sont fournies pour certains dimensions catégoriels, notamment pour le pays de naissance (classification des zones géographiques à plusieurs échelles : continents, etc). Afin de montrer l'usage et l'efficacité de la représentation focus+contexte et de la navigation hiérarchique, nous sélectionnons principalement des dimensions avec ce type de hiérarchie ainsi que des dimensions numériques. Nous nous intéressons à l'ensemble des individus de l'échantillon et sélectionnons les huit dimensions suivant : le pays de naissance

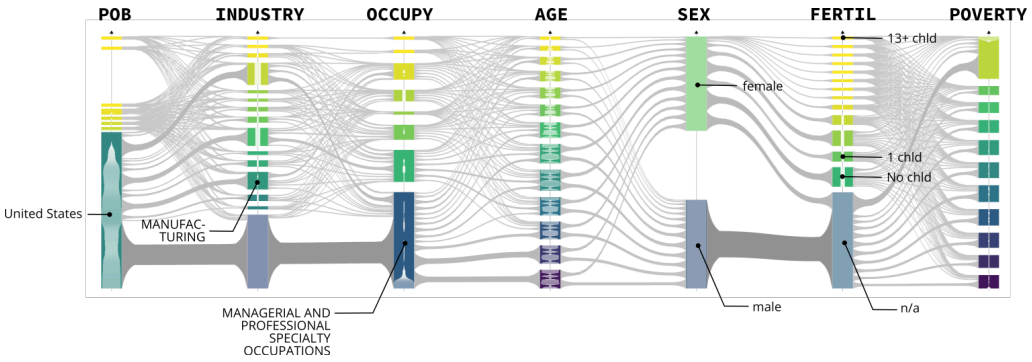


FIGURE 5.35 – Vue initiale 2,50 millions d'individus et 8 dimensions.

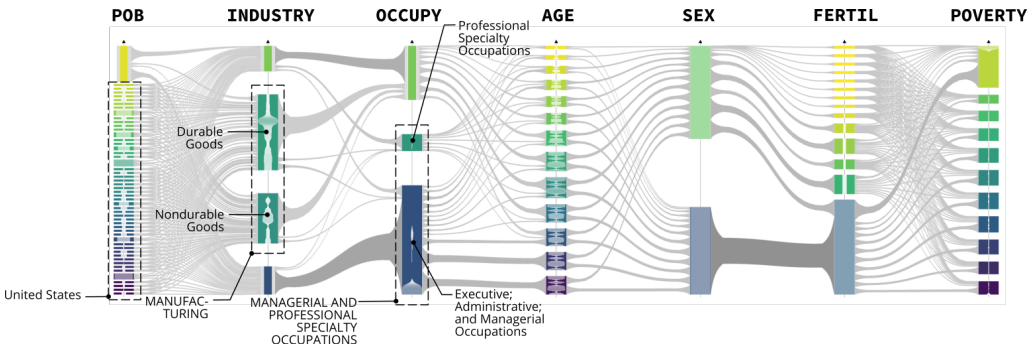


FIGURE 5.36 – Vue résultant de trois ouvertures sur les trois premiers axes, respectivement sur « United States » (POB), « Manufacturing » (INDUSTRY) et « Managerial and professional specialty occupations » (OCCUPY).

(POB), le secteur (INDUSTRY) et l'intitulé de leur poste (OCCUPY) s'ils travaillent, leur âge (AGE), leur genre (SEX), leur nombre d'enfants (FERTIL), et leur niveau de pauvreté (POVERTY).

Dans ce cas d'étude, nous nous proposons d'étudier la relation entre des emplois, selon les dimensions OCCUPY et INDUSTRY, et des données démographiques sur le modèle d'une étude menée par VOSOUGH et al. [161] sur un jeu de données similaire. La figure 5.35 présente la vue initiale : les dimensions POB, INDUSTRY, et OCCUPY sont agrégées selon les hiérarchies fournies. Les catégories SEX et FERTIL sont plates, avec deux valeurs pour SEX, et 14 valeurs pour FERTIL dont la valeur la plus représentée est une valeur de substitution (n/a) pour les hommes. Les dimensions AGE et POVERTY sont numériques et regroupées hiérarchiquement de façon à obtenir une arité de *15* pour les arbres résultants. La vue initiale présente une vue générale de la distribution des données sur chaque dimension ainsi qu'entre les paires de dimensions représentées, avec la hauteur des nœuds et l'épaisseur des arêtes codant pour le nombre d'individus inclus dans l'agrégat. La vue initiale montre immédiatement que la plupart des individus de l'échantillon sont nés aux États-Unis, représentés par un nœud par état, ce qui n'est pas surprenant. On remarque que l'échantillon semble avoir été conçu de manière à inclure autant de femmes que d'hommes, et ce également au sein de chaque catégorie d'âge comme le suggère l'allure des arêtes entre les dimensions SEX et AGE.

Sur la figure 5.36, nous avons augmenté le détail sur les trois premiers axes, respectivement sur les individus nés au États-Unis, le secteur de la confection de produits et les professions de directions et de spécialistes. Cela permet de sélectionner le lien correspondant aux individus occupant des positions de spécialité professionnelle dans la confection de biens durables. La figure 5.37 représente le résultat de cette sélection qui couvre une très faible portion des individus de l'échantillon. Lorsque les sélections sont de faible taille par rapport aux nœuds et liens de la représentation, la jauge montrant la proportion de chaque nœuds/liens couverte par la sélection est peu visible. Dans ce cas, le survol des nœuds et liens donne accès aux valeurs numériques correspondant aux jauges.

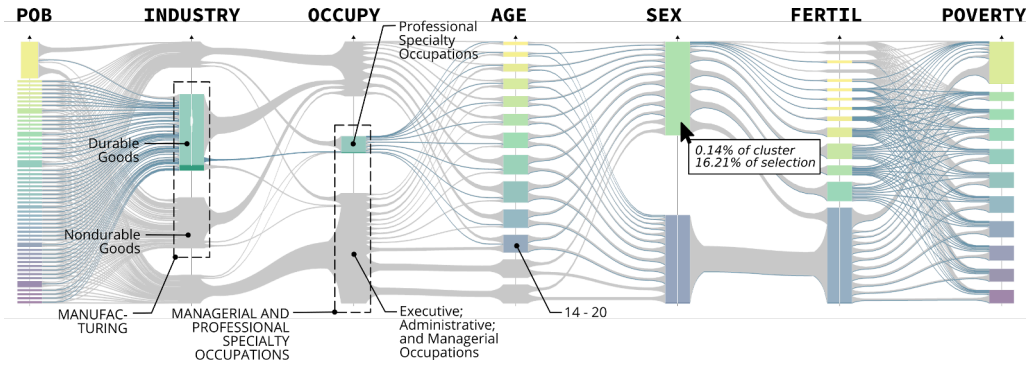


FIGURE 5.37 – Sélection des individus occupant des postes de spécialistes dans le secteur de la confection de biens durables par clic sur un lien entre les dimensions INDUSTRY et OCCUPY. Le survol du nœud pour le genre féminin indique qu'environ 16 % de ses individus sont des femmes.

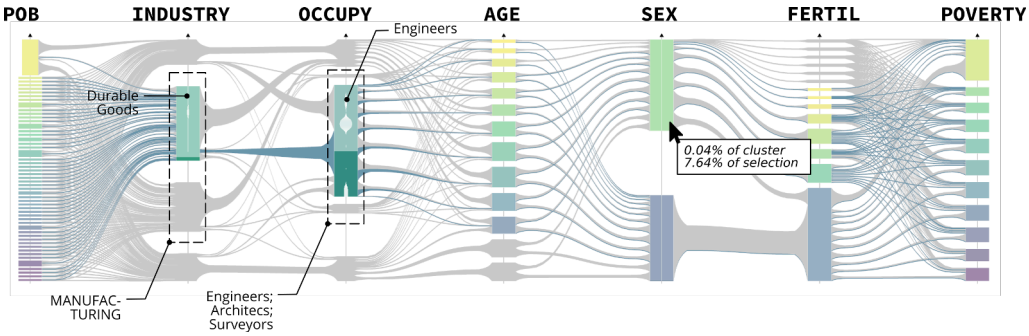


FIGURE 5.38 – Sélection des ingénieurs dans le secteur de la confection de biens durables. Le survol du nœud pour le genre féminin indique qu'environ 8 % de ces individus sont des femmes.

La figure 5.37 montre que les individus sélectionnés ont plus de 14 ans, ce qui est cohérent avec le fait qu'ils travaillent. On peut aussi voir que cette population est constituée d'environ *16,21* % de femmes.

Après quelques étapes de *drill-down* supplémentaires sur l'axe OCCUPY, le focus est établi sur la catégorie des ingénieurs, architectes et géomètres-experts. Nous sélectionnons les individus occupant des postes d'ingénieur dans le domaine de confection de biens durables en cliquant sur le lien connectant les deux nœuds correspondant ce qui résulte en la vue de la figure 5.38. Les femmes représentent *7,64* % de cette sélection comme indique par le survol sur le nœud correspondant et n'ont pas plus de six enfants à la différence de celle de la catégorie professionnelle parente dans la hiérarchie, représentée sur la figure précédente (cf. figure 5.37).

En raffinant encore la sélection aux femmes exclusivement, on peut noter que la plupart ont entre 21 et 34 ans et que la moitié n'ont pas d'enfants (figure 5.39). La faible portion des femmes dans ce secteur est aussi visible par la couleur du lien connectant le nœud de confection de biens durables et celui des ingénieurs. En comparant cette vue et la vue précédente, on peut également voir que certains hommes de cette catégorie professionnelle sont plus âgés que ses femmes : en effet, 60 % des femmes de cette sélection ont entre 21 et 34 ans.

5.6 Conclusion

Dans ce chapitre, nous avons proposé une approche combinant l'agrégation et le calcul des interactions sur une infrastructure distribuée pour répondre au besoin de scalabilité visuelle et computationnelle dans l'exploration de grands jeux de données multi-dimensionnelles.

Nous avons introduit un formalisme utilisant un graphe pour décrire les représentations de coordonnées parallèles abstraites et leurs interactions. Le formalisme permet d'exprimer

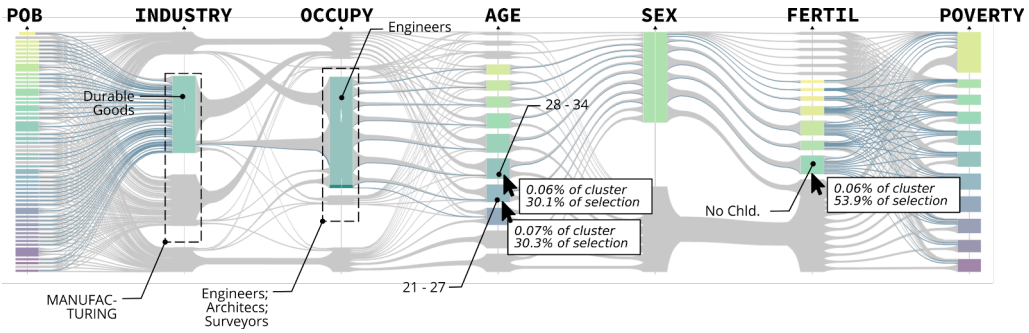


FIGURE 5.39 – Sélection des femmes ingénieures dans le domaine de la confection de produits durables. Plus de 50 % de ces femmes n'ont pas d'enfants.

plusieurs formes d'agrégation et d'interactions existantes et d'évaluer le nombre d'états qu'elles induisent ainsi que la taille des abstractions.

Pour remédier aux limites des représentations abstraites contraignant l'analyse à un unique niveau de granularité, nous avons proposé une interaction de navigation hiérarchique associée à une vue focus+contexte qui utilise l'agrégation du contexte pour maintenir un nombre borné d'éléments visuels et la déformation spatiale pour préserver la visibilité des niveaux de détail les plus fins, ce quel que soit le niveau d'exploration. Ces deux techniques permettent l'exploration des données jusqu'au niveau de détail le plus fin, celui des entités.

Les deux systèmes présentés (avec et sans navigation hiérarchique) couvrent plusieurs techniques pour le passage à l'échelle computationnel : le pré-calcul, le calcul à la volée pas un système d'indexation distribuée (Elasticsearch) et par une application ad hoc (Spark), ainsi que le calcul parallèle. Leurs performances ont été évaluées en termes d'efficacité sur une plateforme d'une quinzaine de machines et en termes de scalabilité pour montrer les possibilités de réduction de latence supplémentaire par l'ajout d'unités de calcul.

Ces deux systèmes utilisent une notion de système *budget*, c.-à-d. borne supérieure, en nombre d'entités visuelles pour n'importe quelle vue résultant de n'importe quelle interaction. Cette borne dépend dans notre implémentation du paramètre k choisi en amont, avant l'exploration et utilisé pour initialiser les hiérarchies calculées automatiquement de telle sorte qu'elles soient k-aire. Pré-calculer ces hiérarchies à l'avantage de fournir un support de navigation initial pour l'utilisateur ; cependant ce support est aussi limitant, notamment car la vue focus+contexte restreint chaque dimension à ne représenter que des nœuds frères comme nœuds *focus*. Les interactions d'édition des hiérarchies sont une manière de remédier à ces restrictions puisqu'elles permettent à l'utilisateur de modifier l'arité des hiérarchies et de réorganiser les fraternités si certaines sont séparées par exemple. Nous pouvons supposer qu'à travers ces interactions d'édition, un utilisateur ne modifierait pas l'arité, initialement choisie raisonnablement en fonction de la bande passante du réseau, jusqu'à en changer l'ordre de grandeur.

6 Conclusion & perspectives

Ce dernier chapitre dresse un bilan des contributions de cette thèse, discute de leurs améliorations et propose quelques perspectives de recherche futures.

6.1 Bilan des contributions

Dans cette thèse, nous nous sommes intéressés à la conception de représentations et interactions scalables pour permettre l'exploration de jeux de données de grande taille en temps interactif dans trois contextes différents : les vues multiples liées pour l'analyse de données complexes, les cartes de densité pour les ensembles de points et les coordonnées parallèles abstraites pour les données multi-dimensionnelles. En particulier, nos travaux se sont articulés autour des deux problématiques de scalabilité visuelle et computationnelle.

Dans le chapitre 3, nous nous sommes intéressés à la conception d'une technique scalable de mise en évidence d'entités dans des vues liées. L'efficacité des approches existantes, par surbrillance ou liens visuels, diminue lorsque les données sont de grande taille alors que les approches existantes utilisant la déformation spatiale sont limitées en application à des vues utilisant la même forme de représentation. Nous avons proposé une technique nommée CorFish qui applique une déformation spatiale élargissant l'espace autour des entités désignées d'intérêt sur de multiples domaines de coordonnées, la rendant ainsi compatible avec de multiples formes de représentation visuelle. Cette technique préserve les bordures des vues et est compatible avec une définition non-binaire de l'intérêt sur les entités. À travers plusieurs exemples et une implémentation prototype, nous avons montré que CorFish peut être appliquée combinée à la surbrillance pour la mise en évidence interactive dans des vues liées et présentant un grand nombre d'entités.

Dans le chapitre 4, nous nous sommes intéressés à la réduction des données nécessaires à la reconstruction de cartes de densité de points. Cette problématique concerne la scalabilité computationnelle de techniques l'exploration de cartes de densité d'un ensemble du plan. Nous avons proposé une approche de réduction géométrique pour répondre au goulot d'étranglement du transfert réseau et de la capacité mémoire du client dans un contexte d'exploration distante de données massives. L'approche consiste en un cadre d'abstraction géométrique appelé CoReDeM similaire à une compression géométrique avec pertes : il prend en entrée un budget de stockage et impose des contraintes *conservatrices* sur les cartes de densité reconstruites. Nous avons procédé à l'évaluation de plusieurs stratégies respectant ce cadre pour identifier les plus efficaces, c.-à-d. celles qui résultent, à fort taux de réduction, en le moins de dégradation de l'apparence de la carte de densité. Les résultats d'une évaluation par métriques à grande échelle et d'une évaluation utilisateur sur cinq stratégies ont identifié les stratégies de *binning* standards et les stratégies utilisant un partitionnement (canopée ou k-moyennes) combiné à une étape de rognage utilisant un diagramme Voronoï comme les plus efficaces.

Enfin, dans le chapitre 5, nous avons présenté un système pour l'exploration de grands jeux de données multi-dimensionnelles qui répond aux deux problématiques de scalabilité visuelle et computationnelle en utilisant une représentation abstraite respectant un budget d'entités visuelles combinée au calcul distribué des interactions de navigation et sélection. Pour permettre le passage à l'échelle visuel, l'interaction de navigation hiérarchique est combinée à une vue focus+contexte où le contexte est agrégé et compressé pour accorder de la visibilité au focus. Nous avons proposé un formalisme sous forme de graphe et de graphe quotient pour relier les données multi-dimensionnelles à la vue abstraite et ses différents états résultant d'interactions de sélection, ré-ordonnement et navigation hiérarchique (drill-down, roll-up). Les systèmes

implémentés démontrent une bonne scalabilité et atteignent des latences d'interaction sous la seconde pour des jeux de données de plusieurs millions d'entités et une quinzaine de machines.

6.2 Perspectives

Les travaux présentés dans cette thèse utilisent des techniques de déformation spatiale, agrégation et calcul distribué pour répondre aux problématiques de scalabilité, visuelle ou computationnelle, dans différents contextes. Notamment, nous avons utilisé les vues focus+contexte sous la forme de déformation spatiale localisée dans CorFish ou d'agrégation à différents nivaux de granularité dans HiePaCo ; et les représentations abstraites dans HiePaCo et CoReDeM dans le but de permettre la visualisation de données massives. Nous présentons ici les perspectives d'amélioration et de recherche pour les trois travaux.

Déformation coordonnées pour les vues multiples (CorFish)

Une question importante à traiter pour CorFish est la comparaison de ses performances d'utilisation par rapport à la mise en évidence par la couleur et les liens visuels dans plusieurs contextes de densité, de périphérique (définition d'écran et distance d'utilisation) et d'échelles de données de façon à identifier les contextes où l'approche de CorFish est bénéfique.

Une seconde perspective de recherche est l'extension de la technique de déformation pour permettre la définition d'une déformation 2D qui serait avantageuse pour réduire les artefacts dans les vues nœuds-liens et pour permettre la déformation de vues aux formes géométriques non-alignées avec les axes de coordonnées comme les *bubble treemap*. Pour cela, une réflexion est à mener sur les propriétés géométriques à conserver de manière analogue aux propriétés définies pour la déformation 1D.

Compression géométriques pour les cartes densité (CoReDeM)

Dans le chapitre 4, nous avons présenté deux approches de comparaison des différentes stratégies de compression implémentées. Les stratégies de compression ayant été comparées par deux méthodes, par des scores de similarité utilisant SSIM et par des classements de similarité établis par des individus. L'évaluation par métrique a fait ressortir sept stratégies prometteuses dont seules quatre ont été comparées entre elles lors de l'évaluation utilisateur. Une évaluation utilisateur complémentaire pourrait permettre d'établir des recommandations d'utilisation sur une base plus équilibrée.

Un argument important de l'approche géométrique proposée relativement au pré-calcul de pyramides d'images est la faible interactivité offerte par une image. Une perspective intéressante pour la méthode générale de CoReDeM serait d'évaluer les possibilités d'implémentation du brossage et lien entre plusieurs vues de cartes de densités, par exemple sous la forme d'une matrice de nuages de points pour des donnés multi-dimensionnelles. Cette perspective comporte deux volets : d'une part la recherche d'une représentation adaptée de la sélection et d'autre part l'implémentation du calcul du lien entre les vues. Les systèmes ou structures de données existants destinés à réduire les latences de brossage et lien entre des vues agrégées par *binning* ne seraient donc pas directement applicables aux stratégies utilisant des formes circulaires, polygonales ou des formes se chevauchant.

Enfin, plusieurs travaux s'intéressent à la simplification visuelle de cartes de densité de points *multi-classe*, c'est-à-dire où la représentation finale transcrit l'appartenance de chaque point des données à une catégorie. L'approche de CoReDeM pourrait être adaptée à cette problématique en superposant ou combinant les cartes de densité simplifiées de chaque classe en considérant par exemple dans un premier temps les différentes techniques proposées par Jo et al. [87].

Coordonnées parallèles hiérarchiques (HIEPACO)

Dans le chapitre 5, nous avons montré l'exploitation des hiérarchies dimensionnelles pour permettre l'exploration des données dans une vue focus+contexte. Dans le cas où les données ne fournissent pas de hiérarchisation sémantique, des hiérarchies sont calculées à partir des données au préalable. La structure de ces hiérarchies peut paraître limitante lors de l'exploration. Une manière de pallier cette limitation est d'augmenter l'outil de visualisation d'interactions d'édition de la hiérarchie qui éditent la partition présentée à l'utilisateur par manipulation directe des méta-nœuds de la vue courante. L'idée est de fournir une manière de corriger les défauts identifiés par l'utilisateur sur la partition qui lui est présentée et d'entériner ces changements sur la hiérarchie, initialement calculée automatiquement, pour autres coupures. Nous avons mené une phase préliminaire de discussion sur deux interactions complémentaires d'édition de la hiérarchie (cf. annexe B) qui pourrait être poursuivie jusqu'à l'intégration au système existant.

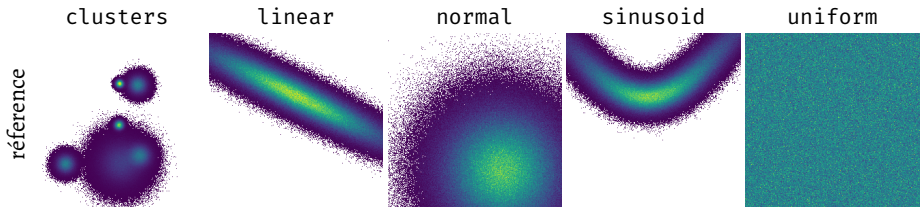
Les approches de traitements des données massives utilisées dans cette thèse s'appuient sur le calcul distribué et le pré-calcul pour accélérer certains traitement et réduire les latences d'interactions. Le pré-calcul a l'avantage de permettre des interactions rapides comme montré dans la version non-hiérarchique des coordonnées parallèles mais possède un coût initial très important : la phase de préparation pour un milliard d'entités et une dizaine de dimensions pouvait prendre plus d'une journée. Le paradigme de visualisation progressive propose une approche alternative au pré-calcul qui consiste à fournir des vues transitoires approximatives, s'améliorant au fur et à mesure que les données sont traitées. Elle a l'avantage de proposer un compromis entre les latences de traitement (pendant l'interaction ou en amont de l'analyse) et la véracité de la visualisation proposée. Pour les coordonnées parallèles, cette approche pourrait être étudiée, notamment combinée au système fonctionnant exclusivement par du calcul à la volée. Ces approches sont notamment intéressantes pour les vues agrégées comme, celle des parallèles abstraites, dont l'apparence peut être conservée relativement stable à mesure qu'elle est calculée sur des échantillons des données de plus en plus grands.

A Données d'évaluation pour CoReDeM

Cette annexe fournit des données complémentaires pour les évaluations menées dans le chapitre 4 (page 45).

Génération des jeux de données synthétiques

Pour chaque jeu de données, les points sont générés dans l'espace $[0, 1] \times [0, 1]$. Chaque point dont les coordonnées sont tirées en dehors de cet espace est régénéré jusqu'à l'obtention de n points. Les distributions de points sont simulées par le générateur de nombres pseudo-aléatoires Mersenne Twister (bibliothèque standard de `C++11`). $\mathcal{N}(\mu, \sigma^2)$ désigne la loi normale de moyenne μ et d'écart-type σ et $\mathcal{U}(a, b)$ désigne la loi uniforme sur l'intervalle $[a, b]$.



`normal` Les n points sont générés avec leur coordonnées tirées suivant $\mathcal{N}(\mu, 0.2^2)$ où μ est tiré suivant $\mathcal{U}(0.2, 0.8)$.

`uniform` Les n points sont générés avec leurs coordonnées tirées suivant $\mathcal{U}(0, 1)$ pour chacune.

`linear` Les valeurs y_1 et y_2 définissant la droite d'équation $y = (y_2 - y_1)x + y_1$ sont générées suivant $\mathcal{U}(0, 1)$. Puis n points de cette droite sont générés aux abscisses tirées suivant $\mathcal{N}(0.5, 0.3^2)$. Pour finir, ces n points sont déplacés selon un bruit aléatoire dont la composante pour chaque direction est tirée suivant $\mathcal{N}(0, 0.05^2)$.

`sinusoid` Une courbe sinusoïdale de la forme $f(x) = s + asin(\frac{x}{p} + s)$ est définie par tirage de s, a, et p suivant $\mathcal{U}(0, 1)$. Puis n points de cette courbe sont générés aux abscisses tirées suivant $\mathcal{N}(0.5, 0.3^2)$. Pour finir, ces n points sont déplacés selon un bruit aléatoire dont la composante pour chaque direction est tirée dans $\mathcal{N}(0, 0.05^2)$.

`clusters` Six centres de cluster sont générés en tirant chacune de leur coordonnées suivant $\mathcal{U}(0.2, 0.8)$. La proportion des n points appartenant à chaque cluster est calculée itérativement en tirant un ratio suivant $\mathcal{U}(0.2, 0.8)$ des points restants, à l'exception du dernier cluster qui récupère les points restants. Puis, m les points du cluster de centre p sont générés suivant $\mathcal{N}(p, (2\frac{m}{n})^2)$.

Évaluation par métriques

La figure A.1 représente les scores SSIM moyens de toutes les stratégies par distribution, pour tous les exemplaires de jeu de données de la distribution et tous les budgets. Les valeurs-p des tests de Friedman pour toutes les données sont fournis sur la figure 4.15 (page 64) et sur la tableau A.1 pour chaque budget. Les valeurs-p des tests statistique de Conover calculant la significativité des comparaisons paire à paire des 34 stratégies dans les matrices des figures 4.15 et 4.16 (pages 64 et 65) sont rapportées pour toutes les données sur le tableau A.2 et par budget dans les tableaux A.3 à A.8. Ces valeurs-p sont ré-ajustées par correction de Bonferroni et sont rapportées orangées lorsque $p < 0,05$.

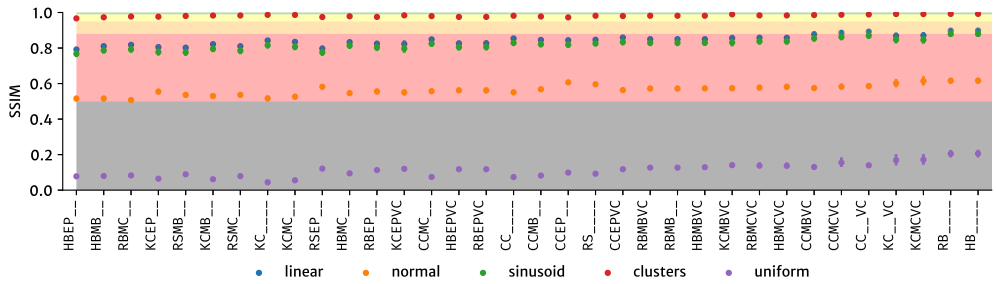


FIGURE A.1 – Score SSIM moyen de chaque stratégie par distribution des données.

k	r (%)	$\chi^2(34)$	valeur-p
655	1	2641, 01	$p \ll 0,008$
1 310	2	2824, 38	$p \ll 0,008$
3 276	5	3316, 89	$p \ll 0,008$
6 553	10	3660, 87	$p \ll 0,008$
13 107	20	3909, 10	$p \ll 0,008$
19 660	30	3979, 69	$p \ll 0,008$

TABLE A.1 – Statistique de Friedman pour chaque budget (correction de Bonferroni : $\alpha = 0,05/8 \approx 0,08$).

stratégie	HBEP	HBMB	RBMC	KCEP	RSMB	KCMB	RSMC	KC	KCMC	RSEP	HBMC	RBEP	KCEPVC	CCMC	HBEPVC	RBEPVC	CC	CCMB	CCEP	RS	CCEPVC	RBMBVC	RBMB	HBMBVC	KCMBVC	RBMCVC	HBMCVC	CCMBVC	CCMCVC	CCVC	KCVC	KCMCVC	RB	HB
HBEP	–	1,00	1,00	0,02	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMB	1,00	–	1,00	1,00	0,83	1,00	0,59	0,11	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMC	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEP	0,02	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,02	0,00	0,97	0,00	0,00	0,27	0,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMB	0,00	0,83	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	1,00	0,07	0,00	1,00	0,00	0,00	0,89	0,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMB	0,01	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,00	1,00	0,04	0,00	1,00	0,00	0,00	0,54	0,10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMC	0,00	0,59	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	1,00	0,11	0,00	1,00	0,00	0,00	1,00	0,25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KC	0,00	0,11	1,00	1,00	1,00	1,00	1,00	–	1,00	0,02	1,00	0,61	0,00	1,00	0,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMC	0,00	0,02	1,00	1,00	1,00	1,00	1,00	1,00	–	0,10	1,00	1,00	0,00	1,00	0,02	0,01	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSEP	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,10	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,48	0,07	0,07	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,09	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEP	0,00	0,00	0,00	0,02	0,07	0,04	0,11	0,61	1,00	1,00	1,00	–	0,87	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,87	–	0,02	1,00	1,00	0,07	0,41	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,43	0,54	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMC	0,00	0,00	0,00	0,97	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,02	–	1,00	1,00	1,00	1,00	0,27	0,07	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	1,00	0,38	0,38	0,14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,44	0,44	0,16	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CC	0,00	0,00	0,00	0,27	0,89	0,54	1,00	1,00	1,00	1,00	1,00	1,00	0,07	1,00	1,00	1,00	–	1,00	0,95	0,29	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMB	0,00	0,00	0,00	0,05	0,17	0,10	0,25	1,00	1,00	1,00	1,00	1,00	0,41	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEP	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,09	1,00	1,00	0,27	1,00	1,00	0,95	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	0,03	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RS	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,02	1,00	1,00	0,07	1,00	1,00	0,29	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	0,11	0,14	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,48	0,00	0,01	1,00	0,00	1,00	1,00	0,00	0,00	1,00	1,00	–	1,00	1,00	1,00	0,37	1,00	1,00	0,12	0,00	0,00	0,00	0,00	0,00	0,00
RBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,07	0,00	0,00	1,00	0,00	0,38	0,44	0,00	0,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,76	0,00	0,00	0,02	0,01	0,00	0,00
RBMB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,07	0,00	0,00	1,00	0,00	0,38	0,44	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,76	0,00	0,00	0,02	0,01	0,00	0,00
HBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	1,00	0,00	0,14	0,16	0,00	0,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,01	0,00	0,06	0,02	0,00	0,00
KCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,37	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00
RBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,03	1,00	1,00	0,00	0,00
HBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,54	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,14	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,02	1,00	1,00	0,00	0,00
CCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,12	0,76	0,76	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	0,00
CCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,02	0,01
CCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,03	0,02	1,00	1,00	–	1,00	1,00	1,00	1,00
KCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,02	0,06	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,00	0,00
KCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,02	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	0,00	0,00
RB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	1,00	0,00	0,00	–	1,00
HB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	1,00	0,00	0,00	1,00	–

TABLE A.2: Valeurs-p du test de Conover pour les résultats d'évaluation par métrique ($\alpha = 0,05$).

stratégie	HBEP	HBMB	RBMC	KCEP	RSMB	KCMB	RSMC	KC	KCMC	RSEP	HBMC	RBEP	KCEPVC	CCMC	HBEPVC	RBEPVC	CC	CCMB	CCEP	RS	CCEPVC	RBMBVC	RBMB	HBMBVC	KCMBVC	RBMCVC	HBMCVC	CCMBVC	CCMCVC	CCVC	KCVC	KCMCVC	RB	HB
HBEP	–	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,06	1,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMB	0,02	–	1,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMC	0,00	1,00	–	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	0,22	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,09	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEP	0,00	1,00	1,00	–	1,00	0,00	1,00	0,00	0,05	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMB	0,00	1,00	1,00	1,00	–	0,00	1,00	0,00	0,01	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMB	0,00	0,00	1,00	0,00	0,00	–	0,02	0,00	1,00	1,00	1,00	0,00	0,00	0,26	0,00	0,00	0,01	0,05	0,14	0,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	0,00	0,00
RSMC	0,00	1,00	1,00	1,00	1,00	0,02	–	0,00	0,21	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,22	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,13	0,04	1,00	1,00	1,00	0,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00	0,00	0,00
KCMC	0,00	0,00	1,00	0,05	0,01	1,00	0,21	0,00	–	1,00	1,00	0,00	0,00	0,03	0,03	0,00	0,00	0,00	0,02	0,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	0,00	0,00
RSEP	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	–	1,00	1,00	0,02	0,00	1,00	1,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMC	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	–	1,00	0,03	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,58	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEP	0,06	1,00	1,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	–	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEPVC	1,00	1,00	0,22	1,00	1,00	0,00	1,00	0,00	0,00	0,02	0,03	1,00	–	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMC	0,00	0,00	0,00	0,00	0,00	0,26	0,00	1,00	0,03	0,00	0,00	0,00	0,00	–	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	0,03	1,00	1,00	0,17	0,00	0,00	1,00	1,00	0,00	0,00
HBEPVC	0,00	1,00	1,00	1,00	1,00	0,00	1,00	0,00	0,03	1,00	1,00	1,00	1,00	0,00	–	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEPVC	0,01	1,00	1,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	1,00	–	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CC	0,00	0,00	0,00	0,00	0,00	0,01	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	–	1,00	1,00	0,03	0,17	1,00	1,00	1,00	0,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00	0,00	0,00
CCMB	0,00	0,00	0,00	0,00	0,00	0,05	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	–	1,00	0,00	0,95	1,00	1,00	1,00	0,00	1,00	1,00	0,80	0,00	0,00	1,00	1,00	0,00	0,00
CCEP	0,00	0,00	0,00	0,00	0,00	0,14	0,00	1,00	0,02	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	–	0,00	1,00	1,00	1,00	1,00	0,01	1,00	1,00	0,30	0,00	0,00	1,00	1,00	0,00	0,00
RS	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	0,00	0,00	–	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00
CCEPVC	0,00	0,00	0,09	0,00	0,00	1,00	0,00	0,04	1,00	1,00	0,58	0,00	1,00	0,00	0,00	0,00	0,17	0,95	1,00	0,00	–	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	0,00	0,00
RBMBVC	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	1,00	1,00	0,00	0,00
RBMB	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	0,00	0,00	1,00	1,00	0,00	0,00
HBMBVC	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,00	0,00	0,00	1,00	1,00	0,00	0,00
KCMBVC	0,00	0,00	1,00	0,05	0,01	1,00	0,22	0,00	1,00	1,00	1,00	0,00	0,00	0,03	0,03	0,00	0,00	0,00	0,01	0,00	1,00	1,00	1,00	1,00	–	0,00	0,00	0,00	0,00	0,00	1,00	1,00	0,00	0,00
RBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	0,00	–	1,00	1,00	0,07	0,00	1,00	1,00	0,00	0,00
HBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	0,00	1,00	–	1,00	0,11	0,00	1,00	1,00	0,00	0,00
CCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,17	0,00	0,00	1,00	0,80	0,30	1,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	–	1,00	0,95	0,00	0,00	0,00	0,00
CCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,07	0,11	1,00	–	1,00	0,00	0,00	0,07	0,15
CCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,95	1,00	–	0,00	0,00	1,00	1,00
KCVC	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	–	1,00	0,00	0,00
KCMCVC	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	1,00	–	0,00	0,00
RB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,07	1,00	0,00	0,00	–	1,00
HB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	1,00	0,00	0,00	1,00	–

TABLE A.3: Valeurs-p du test de Conover pour les résultats d'évaluation par métrique ($k = 655$, $\alpha = 0,05$).

stratégie	HBEP	HBMB	RBMC	KCEP	RSMB	KCMB	RSMC	KC	KCMC	RSEP	HBMC	RBEP	KCEPVC	CCMC	HBEPVC	RBEPVC	CC	CCMB	CCEP	RS	CCEPVC	RBMBVC	RBMB	HBMBVC	KCMBVC	RBMCVC	HBMCVC	CCMBVC	CCMCVC	CCVC	KCVC	KCMCVC	RB	HB
HBEP	–	1,00	0,08	0,00	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMB	1,00	–	1,00	0,82	1,00	0,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	0,50	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMC	0,08	1,00	–	1,00	1,00	0,01	1,00	0,00	0,18	0,02	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEP	0,00	0,82	1,00	–	1,00	1,00	1,00	0,08	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMB	0,03	1,00	1,00	1,00	–	0,02	1,00	0,00	0,40	0,05	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMB	0,00	0,00	0,01	1,00	0,02	–	1,00	1,00	1,00	1,00	1,00	1,00	0,04	1,00	0,88	1,00	0,01	0,00	0,00	0,00	1,00	1,00	1,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMC	0,00	1,00	1,00	1,00	1,00	1,00	–	0,01	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KC	0,00	0,00	0,00	0,08	0,00	1,00	0,01	–	1,00	1,00	0,04	0,00	0,00	1,00	0,00	0,14	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMC	0,00	0,00	0,18	1,00	0,40	1,00	1,00	1,00	–	1,00	1,00	1,00	0,84	0,09	1,00	1,00	0,00	0,00	0,00	0,00	1,00	0,58	0,58	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSEP	0,00	0,00	0,02	1,00	0,05	1,00	1,00	1,00	1,00	–	1,00	1,00	0,11	0,67	1,00	1,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMC	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,04	1,00	1,00	–	1,00	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEP	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	–	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEPVC	0,01	1,00	1,00	1,00	1,00	0,04	1,00	0,00	0,84	0,11	1,00	1,00	–	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMC	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	0,09	0,67	0,00	0,00	0,00	–	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00
HBEPVC	0,00	1,00	1,00	1,00	1,00	0,88	1,00	0,00	1,00	1,00	1,00	1,00	1,00	0,00	–	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEPVC	0,00	0,50	1,00	1,00	1,00	1,00	1,00	0,14	1,00	1,00	1,00	1,00	1,00	0,00	1,00	–	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CC	0,00	0,00	0,00	0,00	0,00	0,01	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	–	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	0,13	1,00	0,00	0,00	0,00	0,11	0,00	0,00	0,00
CCMB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	–	1,00	0,00	1,00	1,00	1,00	1,00	1,00	0,43	1,00	0,00	0,00	0,00	1,00	0,37	0,00	0,00
CCEP	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	–	0,00	1,00	1,00	1,00	1,00	1,00	0,30	1,00	0,00	0,00	0,00	1,00	0,25	0,00	0,00
RS	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00
CCEPVC	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00	1,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	–	1,00	1,00	1,00	1,00	0,00	0,06	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMBVC	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	0,58	1,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	–	1,00	1,00	1,00	0,00	0,23	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMB	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	0,58	1,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	–	1,00	1,00	0,00	0,23	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMBVC	0,00	0,00	0,00	0,00	0,00	0,01	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	–	1,00	0,20	1,00	0,00	0,00	0,00	1,00	0,17	0,00	0,00
KCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	–	0,30	1,00	0,00	0,00	0,00	1,00	0,26	0,00	0,00
RBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,13	0,43	0,30	1,00	0,00	0,00	0,00	0,20	0,30	–	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00
HBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,06	0,23	0,23	1,00	1,00	1,00	–	1,00	0,00	0,00	1,00	1,00	0,00	0,00
CCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	0,00
CCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	–	1,00	0,44	1,00	0,18	0,11
CCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	–	0,00	0,00	1,00	1,00
KCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,44	0,00	–	1,00	0,00	0,00	0,00
KCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	0,37	0,25	1,00	0,00	0,00	0,00	0,17	0,26	1,00	1,00	1,00	1,00	0,00	1,00	–	0,00	0,00
RB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,18	1,00	0,00	0,00	–	1,00
HB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	1,00	0,00	0,00	1,00	–

TABLE A.4: Valeurs-p du test de Conover pour les résultats d'évaluation par métrique ($k = 1310$, $\alpha = 0,05$).

stratégie	HBEP	HBMB	RBMC	KCEP	RSMB	KCMB	RSMC	KC	KCMC	RSEP	HBMC	RBEP	KCEPVC	CCMC	HBEPVC	RBEPVC	CC	CCMB	CCEP	RS	CCEPVC	RBMBVC	RBMB	HBMBVC	KCMBVC	RBMCVC	HBMCVC	CCMBVC	CCMCVC	CCVC	KCVC	KCMCVC	RB	HB
HBEP	–	1,00	1,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMB	1,00	–	1,00	0,00	0,58	0,19	0,08	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMC	1,00	1,00	–	0,00	1,00	1,00	0,51	0,06	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEP	0,00	0,00	0,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMB	0,01	0,58	1,00	1,00	–	1,00	1,00	1,00	1,00	0,02	1,00	1,00	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMB	0,00	0,19	1,00	1,00	1,00	–	1,00	1,00	1,00	0,07	1,00	1,00	0,00	1,00	1,00	1,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMC	0,00	0,08	0,51	1,00	1,00	1,00	–	1,00	1,00	0,17	1,00	1,00	0,00	1,00	1,00	1,00	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KC	0,00	0,01	0,06	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,01	1,00	1,00	1,00	0,16	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSEP	0,00	0,00	0,00	1,00	0,02	0,07	0,17	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,24	0,00	0,15	0,02	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,06	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEP	0,00	0,00	0,01	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	0,06	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEPVC	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,01	1,00	1,00	0,06	0,06	–	1,00	0,07	1,00	1,00	1,00	1,00	0,01	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBEPVC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,07	1,00	–	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEPVC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CC	0,00	0,00	0,00	1,00	0,00	0,01	0,02	0,16	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	1,00	0,22	0,22	0,14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMB	0,00	0,00	0,00	1,00	0,00	0,01	0,02	0,15	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,00	1,00	0,23	0,23	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEP	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,24	0,00	0,00	1,00	0,00	0,00	0,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RS	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	1,00	–	1,00	1,00	1,00	1,00	0,12	1,00	1,00	0,21	0,00	0,00	0,00	0,00	0,00	0,00
CCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,00	0,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,00	0,04	0,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	1,00	0,00	0,00	0,00	0,22	0,23	1,00	1,00	1,00	–	1,00	1,00	0,00	0,28	0,30	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	1,00	0,00	0,00	0,00	0,22	0,23	1,00	1,00	1,00	1,00	–	1,00	0,00	0,28	0,30	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	1,00	0,00	0,00	0,00	0,14	0,15	1,00	1,00	1,00	1,00	1,00	–	0,00	0,43	0,46	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,12	0,00	0,00	0,00	0,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00
RBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	1,00	0,04	0,28	0,28	0,43	1,00	–	1,00	1,00	1,00	0,00	1,00	1,00	0,00
HBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03	1,00	0,05	0,30	0,30	0,46	1,00	1,00	–	1,00	1,00	0,00	1,00	1,00	0,00
CCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,21	0,00	0,00	0,00	0,00	1,00	1,00	1,00	–	1,00	0,80	1,00	1,00	0,00	0,00
CCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,75	0,77
CCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,80	1,00	–	1,00	1,00	1,00	1,00
KCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,04	0,04
KCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	0,43	0,44
RB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,75	1,00	0,04	0,43	–	1,00
HB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,77	1,00	0,04	0,44	1,00	–

TABLE A.5: Valeurs-p du test de Conover pour les résultats d'évaluation par métrique ($k = 3276$, $\alpha = 0,05$).

stratégie	HBEP	HBMB	RBMC	KCEP	RSMB	KCMB	RSMC	KC	KCMC	RSEP	HBMC	RBEP	KCEPVC	CCMC	HBEPVC	RBEPVC	CC	CCMB	CCEP	RS	CCEPVC	RBMBVC	RBMB	HBMBVC	KCMBVC	RBMCVC	HBMCVC	CCMBVC	CCMCVC	CCVC	KCVC	KCMCVC	RB	HB
HBEP	–	1,00	1,00	0,02	0,01	0,10	0,00	0,06	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMB	1,00	–	1,00	0,11	0,06	0,49	0,01	0,29	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMC	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEP	0,02	0,11	1,00	–	1,00	1,00	1,00	1,00	1,00	0,06	1,00	1,00	0,00	1,00	0,00	0,28	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMB	0,01	0,06	1,00	1,00	–	1,00	1,00	1,00	1,00	0,12	1,00	1,00	0,00	1,00	0,01	0,51	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMB	0,10	0,49	1,00	1,00	1,00	–	1,00	1,00	1,00	0,01	1,00	1,00	0,00	1,00	0,00	0,06	0,46	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMC	0,00	0,01	1,00	1,00	1,00	1,00	–	1,00	1,00	0,92	1,00	1,00	0,00	1,00	0,06	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KC	0,06	0,29	1,00	1,00	1,00	1,00	1,00	–	1,00	0,02	1,00	1,00	0,00	1,00	0,00	0,11	0,77	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSEP	0,00	0,00	0,00	0,06	0,12	0,01	0,92	0,02	1,00	–	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMC	0,00	0,00	0,01	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,00	1,00	1,00	1,00	1,00	1,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEP	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	0,00	1,00	1,00	1,00	1,00	1,00	0,01	0,14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,47	0,00	0,00	0,00	0,00	0,00	0,00
CCMC	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	–	1,00	1,00	1,00	1,00	0,00	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBEPVC	0,00	0,00	0,00	0,00	0,01	0,00	0,06	0,00	1,00	1,00	1,00	1,00	0,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEPVC	0,00	0,00	0,00	0,28	0,51	0,06	1,00	0,11	1,00	1,00	1,00	1,00	0,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CC	0,00	0,00	0,00	1,00	1,00	0,46	1,00	0,77	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	–	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMB	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	–	0,15	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEP	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,01	1,00	0,00	1,00	1,00	1,00	0,15	–	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RS	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	1,00	0,01	0,14	1,00	0,03	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,02	0,00	0,00	0,00	1,00	1,00	–	1,00	1,00	1,00	0,00	0,88	1,00	0,04	0,00	0,00	0,00	0,00	0,00	0,00
RBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,01	0,00	0,00	0,00	1,00	1,00	1,00	–	1,00	1,00	0,00	1,00	1,00	0,11	0,00	0,00	0,00	0,00	0,00	0,00
RBMB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,01	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	0,00	1,00	1,00	0,11	0,00	0,00	0,00	0,00	0,00	0,00
HBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,20	1,00	1,00	1,00	–	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,08	0,01
RBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,88	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	0,02	0,00	0,00	0,00	0,00
HBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	0,01	0,00	0,00	0,00
CCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,47	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,04	0,11	0,11	1,00	1,00	1,00	1,00	–	1,00	0,03	0,59	0,02	0,00	0,00
CCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,14
CCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,03	1,00	–	1,00	1,00	1,00	1,00
KCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,02	0,01	0,59	1,00	1,00	–	1,00	1,00	1,00
KCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,02	1,00	1,00	1,00	–	1,00	1,00
RB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,08	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00
HB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,14	1,00	1,00	1,00	1,00	–

TABLE A.6: Valeurs-p du test de Conover pour les résultats d'évaluation par métrique ($k = 6553$, $\alpha = 0,05$).

stratégie	HBEP	HBMB	RBMC	KCEP	RSMB	KCMB	RSMC	KC	KCMC	RSEP	HBMC	RBEP	KCEPVC	CCMC	HBEPVC	RBEPVC	CC	CCMB	CCEP	RS	CCEPVC	RBMBVC	RBMB	HBMBVC	KCMBVC	RBMCVC	HBMCVC	CCMBVC	CCMCVC	CCVC	KCVC	KCMCVC	RB	HB
HBEP	–	1,00	1,00	1,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMB	1,00	–	1,00	1,00	0,01	0,53	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMC	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,08	1,00	0,00	0,80	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEP	1,00	1,00	1,00	–	1,00	1,00	1,00	0,42	0,00	0,00	0,00	0,00	0,00	0,41	0,00	0,00	0,02	1,00	0,00	0,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMB	0,00	0,01	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMB	0,01	0,53	1,00	1,00	1,00	–	1,00	1,00	0,18	1,00	1,00	0,02	0,00	1,00	0,00	0,00	1,00	1,00	0,03	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMC	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KC	0,00	0,00	1,00	0,42	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMC	0,00	0,00	0,00	0,00	1,00	0,18	1,00	1,00	–	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSEP	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMC	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEP	0,00	0,00	0,00	0,00	1,00	0,02	1,00	1,00	1,00	1,00	1,00	–	0,00	1,00	1,00	1,00	1,00	0,93	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	0,01	0,00	0,00	0,00	0,00	0,00
CCMC	0,00	0,00	1,00	0,41	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	–	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	–	1,00	0,07	0,00	1,00	0,01	1,00	0,47	0,47	0,23	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	1,00	–	0,14	0,00	1,00	0,01	0,90	0,25	0,25	0,12	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CC	0,00	0,00	0,08	0,02	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,07	0,14	–	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMB	0,00	0,01	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,93	0,00	1,00	0,00	0,00	1,00	–	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEP	0,00	0,00	0,00	0,00	1,00	0,03	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	–	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RS	0,00	0,00	0,80	0,17	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,01	0,01	1,00	1,00	1,00	–	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	0,90	0,00	0,00	0,00	0,00	–	1,00	1,00	1,00	0,00	1,00	1,00	0,17	0,00	0,00	0,00	0,00	0,00	0,00
RBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,47	0,25	0,00	0,00	0,00	0,00	1,00	–	1,00	1,00	0,00	1,00	1,00	0,63	0,00	0,00	0,00	0,00	0,00	0,00
RBMB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,47	0,25	0,00	0,00	0,00	0,00	1,00	1,00	–	1,00	0,00	1,00	1,00	0,63	0,00	0,00	0,00	0,00	0,00	0,00
HBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,23	0,12	0,00	0,00	0,00	0,00	1,00	1,00	1,00	–	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	0,05	0,06	0,61	1,00	1,00	1,00	1,00	1,00	1,00
RBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,05	–	1,00	1,00	0,01	0,00	0,00	0,00	0,00	0,00
HBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,06	1,00	–	1,00	0,01	0,00	0,00	0,00	0,00	0,00
CCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,17	0,63	0,63	1,00	0,61	1,00	1,00	–	1,00	0,17	0,00	0,00	0,00	0,00
CCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,86	0,51	0,03
CCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,01	0,01	0,17	1,00	–	1,00	1,00	1,00	1,00
KCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	1,00	–	1,00	1,00	1,00
KCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,86	1,00	1,00	–	1,00	1,00
RB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,51	1,00	1,00	1,00	–	1,00
HB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,03	1,00	1,00	1,00	1,00	–

TABLE A.7: Valeurs-p du test de Conover pour les résultats d'évaluation par métrique ($k = 13107$, $\alpha = 0,05$).

stratégie	HBEP	HBMB	RBMC	KCEP	RSMB	KCMB	RSMC	KC	KCMC	RSEP	HBMC	RBEP	KCEPVC	CCMC	HBEPVC	RBEPVC	CC	CCMB	CCEP	RS	CCEPVC	RBMBVC	RBMB	HBMBVC	KCMBVC	RBMCVC	HBMCVC	CCMBVC	CCMCVC	CCVC	KCVC	KCMCVC	RB	HB
HBEP	–	1,00	0,74	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMB	1,00	–	1,00	1,00	0,13	1,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,84	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBMC	0,74	1,00	–	1,00	1,00	1,00	1,00	0,00	0,00	0,42	0,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00	0,66	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEP	1,00	1,00	1,00	–	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMB	0,00	0,13	1,00	0,00	–	1,00	1,00	1,00	0,25	1,00	1,00	1,00	0,00	1,00	0,00	0,04	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMB	0,00	1,00	1,00	0,01	1,00	–	1,00	1,00	0,02	1,00	0,16	1,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSMC	0,00	0,02	1,00	0,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,04	0,31	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KC	0,00	0,00	0,00	0,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMC	0,00	0,00	0,00	0,00	0,25	0,02	1,00	1,00	–	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	0,03	1,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RSEP	0,00	0,00	0,42	0,00	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMC	0,00	0,00	0,00	0,00	1,00	0,16	1,00	1,00	1,00	1,00	–	1,00	0,00	1,00	1,00	1,00	1,00	0,26	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEP	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	–	0,00	1,00	1,00	1,00	1,00	1,00	1,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	0,17	0,00	0,00	0,00	0,00	0,00
CCMC	0,00	0,01	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	–	0,09	0,70	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
HBEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,04	1,00	1,00	1,00	1,00	1,00	0,00	0,09	–	1,00	0,12	0,00	0,67	0,00	0,35	0,26	0,26	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RBEPVC	0,00	0,00	0,00	0,00	0,04	0,00	0,31	1,00	1,00	1,00	1,00	1,00	0,00	0,70	1,00	–	0,88	0,01	1,00	0,00	0,04	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CC	0,00	0,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,12	0,88	–	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCMB	0,00	0,84	1,00	0,01	1,00	1,00	1,00	1,00	0,03	1,00	0,26	1,00	0,00	1,00	0,00	0,01	1,00	–	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEP	0,00	0,00	0,66	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	0,67	1,00	1,00	1,00	–	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
RS	0,02	1,00	1,00	1,00	1,00	1,00	1,00	0,05	0,00	1,00	0,00	0,01	0,00	1,00	0,00	0,00	1,00	1,00	1,00	–	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
CCEPVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	1,00	0,00	0,35	0,04	0,00	0,00	0,00	0,00	–	1,00	1,00	1,00	0,00	1,00	1,00	0,10	0,00	0,00	0,00	0,00	0,00	0,00
RBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,26	0,03	0,00	0,00	0,00	0,00	1,00	–	1,00	1,00	0,00	1,00	1,00	0,14	0,00	0,00	0,00	0,00	0,00	0,00
RBMB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,26	0,03	0,00	0,00	0,00	0,00	1,00	1,00	–	1,00	0,00	1,00	1,00	0,14	0,00	0,00	0,00	0,00	0,00	0,00
HBMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	–	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00
KCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	–	0,00	0,01	0,16	1,00	1,00	1,00	1,00	1,00	1,00
RBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	–	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00
HBMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	0,01	1,00	–	1,00	1,00	0,02	0,00	0,00	0,00	0,00
CCMBVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,10	0,14	0,14	1,00	0,16	1,00	1,00	–	1,00	0,40	0,00	0,00	0,00	0,00
CCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00	1,00	0,20	1,00	0,21
CCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,02	0,40	1,00	–	1,00	1,00	1,00	1,00
KCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	1,00	–	1,00	1,00	1,00
KCMCVC	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,20	1,00	1,00	–	1,00	1,00
RB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00	–	1,00
HB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,21	1,00	1,00	1,00	1,00	–

TABLE A.8: Valeurs-p du test de Conover pour les résultats d'évaluation par métrique ($k = 19660$, $\alpha = 0,05$).

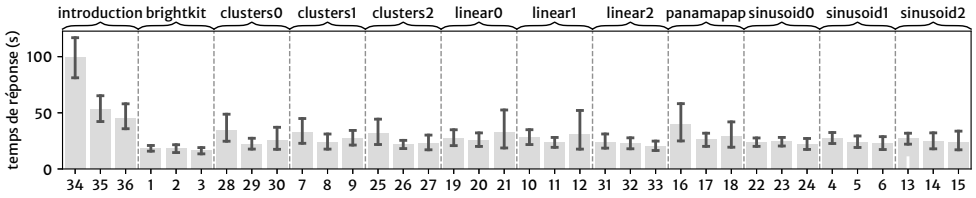


FIGURE A.2 – Temps de réponse moyen des participants par question. Les barres d'erreur correspondent l'intervalle de confiance à 95%.

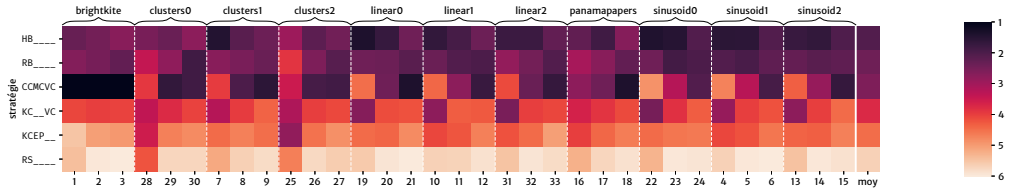


FIGURE A.3 – Classement moyen des stratégies en fonction des questions et en moyenne.

Évaluation utilisateur

La figure A.2 rapporte le temps moyen de réponse des 25 participants aux 33 questions d'évaluation et aux trois questions d'introduction de l'étude. Les questions sont groupées par jeu de données, à l'exception des trois questions d'introduction non comptabilisées dans les analyses. Les trois questions d'un bloc associé à un jeu de données correspondent aux trois budgets dans l'ordre croissant (gauche à droite) sur toutes les figures. La figure A.3 rapporte le score moyen de chaque stratégie par question, c.-à-d. par jeu de données et par budget, ainsi que dans l'ensemble. Enfin, la tableau A.9 rapporte les résultats des tests de Conover pour la comparaison par paire des résultats utilisés pour indiquer la significativité sur la figure 4.20.

stratégie	KCEP	RS	CCMCVC	KCVC	RB	HB
KCEP	–	0,00	0,00	0,00	0,00	0,00
RS	0,00	–	0,00	0,00	0,00	0,00
CCMCVC	0,00	0,00	–	0,00	0,33	0,01
KCVC	0,00	0,00	0,00	–	0,00	0,00
RB	0,00	0,00	0,33	0,00	–	0,09
HB	0,00	0,00	0,01	0,00	0,09	–

(a) Tous budgets confondus, $\alpha = 0,05$.

stratégie	KCEP	RS	CCMCVC	KCVC	RB	HB
KCEP	–	0,01	0,99	0,01	0,00	0,00
RS	0,01	–	0,00	0,00	0,00	0,00
CCMCVC	0,99	0,00	–	0,08	0,01	0,00
KCVC	0,01	0,00	0,08	–	1,00	0,01
RB	0,00	0,00	0,01	1,00	–	0,06
HB	0,00	0,00	0,00	0,01	0,06	–

(b) $k = 1310$, $\alpha = 0,05$.

stratégie	KCEP	RS	CCMCVC	KCVC	RB	HB
KCEP	–	0,00	0,00	0,42	0,00	0,00
RS	0,00	–	0,00	0,00	0,00	0,00
CCMCVC	0,00	0,00	–	0,00	1,00	0,68
KCVC	0,42	0,00	0,00	–	0,00	0,00
RB	0,00	0,00	1,00	0,00	–	0,61
HB	0,00	0,00	0,68	0,00	0,61	–

(c) $k = 6553$, $\alpha = 0,05$.

stratégie	KCEP	RS	CCMCVC	KCVC	RB	HB
KCEP	–	0,01	0,00	0,37	0,00	0,00
RS	0,01	–	0,00	0,00	0,00	0,00
CCMCVC	0,00	0,00	–	0,00	0,48	0,11
KCVC	0,37	0,00	0,00	–	0,00	0,00
RB	0,00	0,00	0,48	0,00	–	1,00
HB	0,00	0,00	0,11	0,00	1,00	–

(d) $k = 19660$, $\alpha = 0,05$.

TABLE A.9 – Valeurs-p du test de Conover pour les résultats d'évaluation utilisateur. Les valeurs-p ont été réajusté par correction de Bonferonni pour $\alpha = 0,05$.

B Édition de hiérarchies dans HiePaCo

En complément aux interactions de navigation hiérarchiques pour les coordonnées parallèles présentées dans le chapitre 5 nous avons étudié deux opérations élémentaires d'édition, la *fusion* et la *fragmentation*, pour permettre la modification de la hiérarchie initialement prédéfinie. Cette annexe présente cette discussion.

La fragmentation consiste à la substitution d'un nœud de la hiérarchie par plusieurs autres. La fusion consiste à la substitution de deux nœuds ou plus par un unique nœud. Similairement aux opérations de *drill-down* (respectivement *roll-up*), une interaction de *fragmentation* (respectivement de *fusion*) aura aussi pour effet de raffiner (respectivement rendre plus grossière) la partition des nœuds et par conséquence la représentation.

Dans les deux cas, la taille du changement incrémental nécessaire à la mise à jour de la vue dépend du nombre de nouveaux nœuds introduits et du nombre de méta-nœuds sur les axes voisins de celui de l'interaction. Pour r nouveaux nœuds ($r = 1$ avec la fusion) et k le nombre maximal de méta-nœuds sur les axes voisins, le nombre de méta-arêtes à transférer est $O(rk)$. Nous présentons plusieurs variantes des opérations de fragmentation et de fusion que l'on compare du point de vue du nombre d'arêtes à modifier. Les figures B.1 et B.2 illustrent les opérations comparées.

Fragmentation

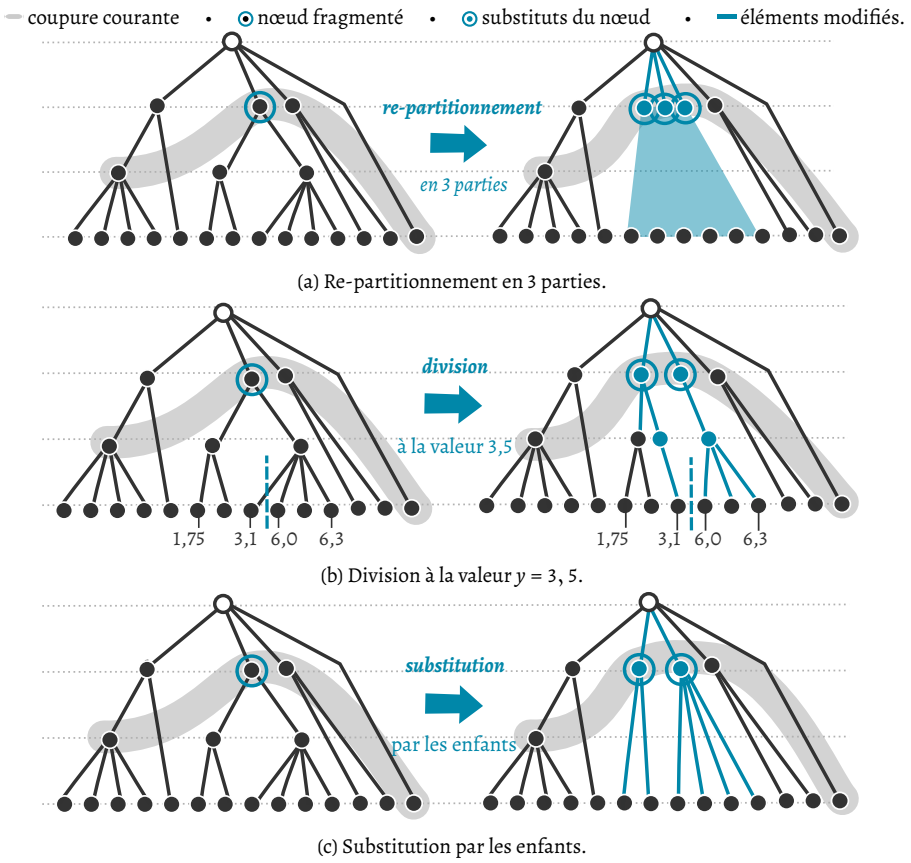
Nous décrivons trois variantes de la fragmentation en commençant par la plus expressive.

Re-partitionnement en r parties

La manière la plus générale de fractionner un nœud est de le diviser en un nombre r de nœuds. Dans le cas général, le sous-arbre du nœud fragmenté doit être remplacé par r sous-arbres tout en préservant le reste de la hiérarchie (cf. figure B.1a). Cela peut être accompli en re-calculant un partitionnement hiérarchique des feuilles couvertes par le nœud fragmenté. Le désavantage de cette approche est que l'obtention automatique d'une hiérarchie pertinente peut être très coûteuse en particulier pour les grands jeux de données, et donc non-faisable en temps interactif.

Division à la valeur y

La seconde manière de fragmenter un nœud est de le remplacer par deux nœuds séparant l'intervalle couvert par le nœud fragmenté en une valeur y, choisie par l'utilisateur . On dit qu'un nœud *couvre un intervalle* $[a, b]$ quand a et b sont les extrema des valeurs des feuilles du nœud. Pour diviser un nœud m en la valeur y, le processus divise m en deux nœuds m_- et m_+ et procède récursivement à la division de chacun des descendants de m qui couvre un intervalle contenant y en commençant par les enfants de m. Les enfants de m couvrant des valeurs supérieures (respectivement inférieure) à y sont assignés à m_+ (respectivement à m_-). En supposant que l'arbre est k-aire et de profondeur h, il y a au plus h nœuds divisés ce qui correspond à $O(kh)$ éditions d'arête, une pour chaque enfant de nœud divisé.



(a) Re-partitionnement en 3 parties.

(b) Division à la valeur $y = 3,5$.

(c) Substitution par les enfants.

FIGURE B.1 – Trois stratégies de fragmentation d'un nœud (modification de la hiérarchie augmentant la granularité).

Substitution par partition des enfants

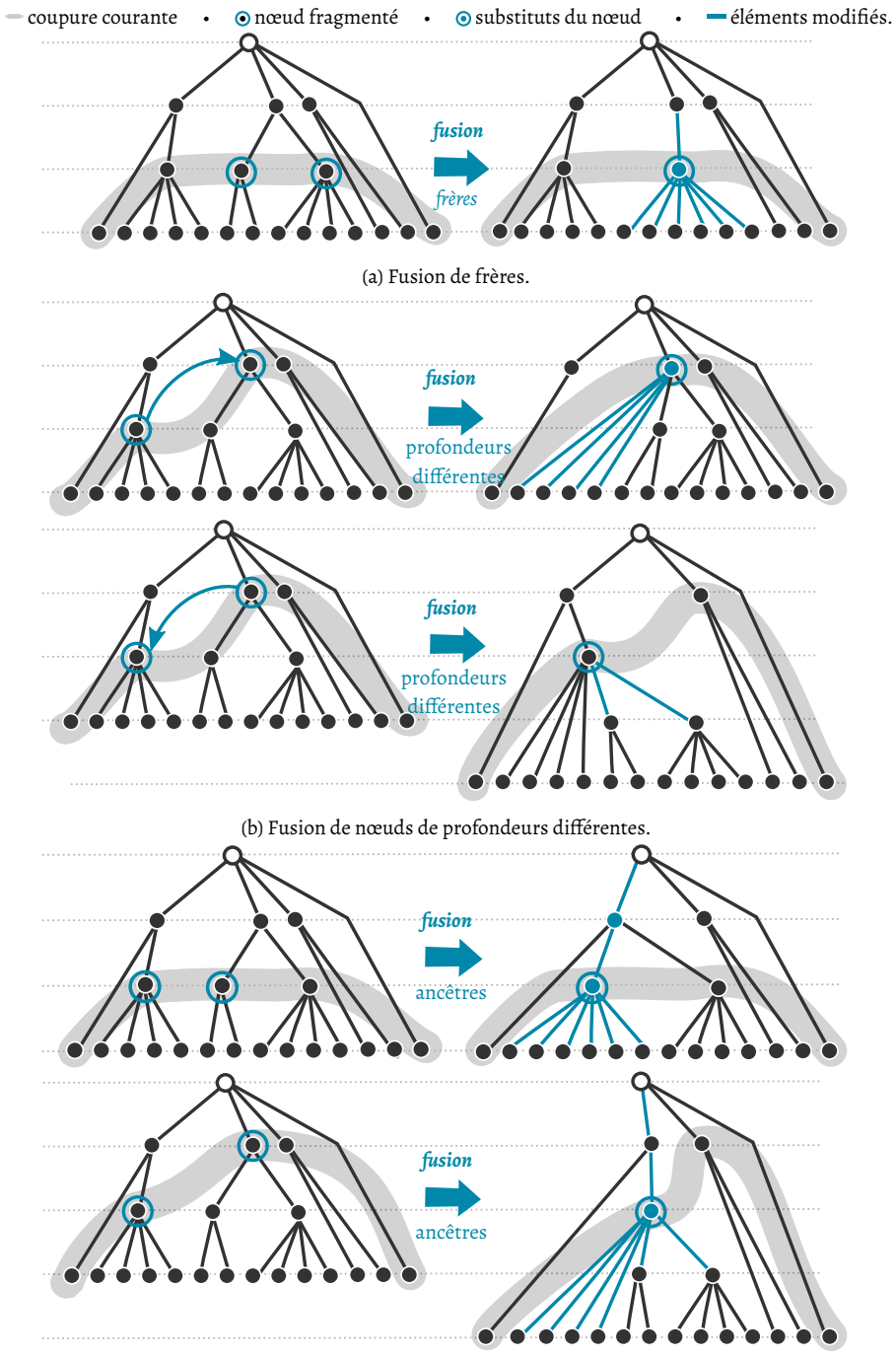
La troisième manière de fragmenter un nœud est de le remplacer par une partition non triviale de ses enfants voire par ses enfants immédiatement comme illustrée sur la figure B.1c. Le pire des cas correspond à ajouter autant de nœuds que le nœud divisé ne possède d'enfants. Pour un arbre k-aire, cela représente $O(k)$ éditions d'arêtes pour connecter les nouveaux nœuds à leur parent et $O(k)$ arêtes pour réassigner les enfants du nœud divisé aux nouveaux nœuds. Cette stratégie est la moins coûteuse des trois. Dans les fait, elle s'apparente à l'opération de *drill-down* sans l'étape de réduction du contexte.

Fusion

Deux nœuds ne peuvent être fusionnés que s'ils couvrent deux ensembles de feuilles consécutifs feuilles de l'abre, c.-à-d. s'il n'existe aucune feuille positionnée entre les deux ensembles. Sans modification supplémentaire, la fusion de deux nœuds non-frères en un unique est impossible car elle invalide la structure arborescente de la hiérarchie. Dans un premier temps nous décrivons l'opération de fusion de nœuds frères puis deux stratégies pour le cas général.

Fusion de frères

La fusion de deux nœuds frères est directe : elle consiste à fusionner les deux nœuds et à réassigner les enfants des deux nœuds fusionnés au nœud résultant (cf. figure B.2a). Cette opération revient à éditer $O(k)$ arêtes pour une hiérarchie k-aire.



(a) Fusion de frères.

(b) Fusion de nœuds de profondeurs différentes.

(c) Fusion des parents.

FIGURE B.2 – Trois stratégies de fusion appliquées à des paires de nœuds. Sur la gauche, ⊙ désigne les nœuds fusionnés sur la gauche et leur substitut. En bleu, les nœuds et arêtes de la hiérarchie modifiée par la fusion. En gris, un exemple de coupure selon laquelle la fusion peut être appliquée.

Fusion de nœuds de profondeurs différentes

Deux nœuds de profondeurs différentes peuvent être fusionnés l'un *dans l'autre* en réassignant le lien de parenté des enfants de l'un des deux à l'autre, ce qui induit $O(k)$ éditions pour un arbre k-aire. Cette stratégie a l'avantage d'introduire peu de changement à la structure de la hiérarchie mais elle n'est pas commutative (cf. figure B.2b) et le choix de la direction d'application n'est pas

évidente. Si les deux nœuds sont de profondeurs différentes, alors l'arbre modifié par leur fusion peut être plus profond que l'arbre original. La fusion de deux nœuds est un cas particulier de cette approche.

Fusion des parents

Une seconde stratégie pour fusionner des nœuds n'étant pas frères est de fusionner leurs ancêtres, et plus précisément la chaîne de nœuds connectant leur parent à la racine. En pratique, deux nœuds peuvent partager des ancêtres, ainsi les ancêtres ne sont fusionnés que depuis leur parent jusqu'à leur ancêtre commun le plus profond. Contrairement à la stratégie précédente, cette opération est commutative. En particulier, lorsque les deux nœuds à fusionner ont la même profondeur le nombre de paires d'ancêtres à fusionner est le même (voir figure B.2c, en haut). Dans le cas général, les enfants du moins profond des deux nœuds à fusionner se voient changer de profondeur en recevant la part d'ancêtres du nœud le plus profond. Cela résulte en un changement de la hauteur de l'arbre (voir la figure B.2c, en bas). Le coût de la fusion des ancêtres dépend de la profondeur de l'arbre. Pour un arbre k-aire de hauteur h, au plus h paires d'ancêtres sont fusionnées. Cela corresponds à $O(kh)$ éditions d'arêtes, une pour chaque enfant d'un nœud fusionné. L'opération de fusion des enfants peut être vue comme un cas particulier de celle-ci.

Bibliographie

[1] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden et Ion Stoica. « BlinkDB : queries with bounded errors and bounded response times on very large data ». In : *Eighth Eurosys Conference 2013, EuroSys '13, Prague, Czech Republic, April 14-17, 2013*. 2013, p. 29-42. DOI : 10.1145/2465351.2465355 (cf. p. 21).

[2] Akamai. *Connectivity Report (Q1 2017) Volume 10 / Number 1*. 2017. URL : https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q1-2017-state-of-the-internet-connectivity-report.pdf (cf. p. 16).

[3] Caroline Appert et Michel Beaudouin-Lafon. « *SwingStates* : adding state machines to Java and the Swing toolkit ». In : *Softw., Pract. Exper.* 38.11 (2008), p. 1149-1182. DOI : 10.1002/spe.867 (cf. p. 11).

[4] M. Asch et al. « Big data and extreme-scale computing ». In : *IJHPCA* 32.4 (2018), p. 435-479. DOI : 10.1177/1094342018778123 (cf. p. 2).

[5] Sven Bachthaler et Daniel Weiskopf. « Continuous Scatterplots ». In : *IEEE Trans. Vis. Comput. Graph.* 14.6 (2008), p. 1428-1435. DOI : 10.1109/TVCG.2008.119 (cf. p. 19, 48).

[6] Michelle Q. Wang Baldonado, Allison Woodruff et Allan Kuchinsky. « Guidelines for using multiple views in information visualization ». In : *Proceedings of the working conference on Advanced visual interfaces - AVI '00*. 2000, p. 110-119. ISBN : 1-58113-252-2. DOI : 10.1145/345513.345271 (cf. p. 11, 24-26).

[7] Lyn Bartram, Albert Ho, John Dill et Frank Henigman. « The Continuous Zoom : A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces ». In : *Proceedings of the 8th Annual ACM Symposium on User Interface Software and Technology, UIST 1995, Pittsburgh, PA, USA, November 14-17, 1995*. 1995, p. 207-215. DOI : 10.1145/215585.215977 (cf. p. 25).

[8] Leilani Battle, Remco Chang et Michael Stonebraker. « Dynamic Prefetching of Data Tiles for Interactive Visualization ». In : *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*. 2016, p. 1363-1375. DOI : 10.1145/2882903.2882919 (cf. p. 20).

[9] Richard A. Becker et William S. Cleveland. « Brushing Scatterplots ». In : *Technometrics* 29.2 (1987), p. 127-142. ISSN : 0040-1706. DOI : 10.2307/1269768 (cf. p. 8, 10, 24, 26, 28).

[10] Richard A. Becker, William S. Cleveland et Allan R. Wilks. « Dynamic Graphics for Data Analysis ». In : *Statistical Science* 2.4 (1987), p. 355-383. ISSN : 0883-4237. URL : http://www.jstor.org/stable/2245523 (cf. p. 8).

[11] Benjamin B. Bederson et James D. Hollan. « Pad++ : A Zooming Graphical Interface for Exploring Alternate Interface Physics ». In : *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology, UIST 1994, Marina del Rey, CA, USA, November 2-4, 1994*. 1994, p. 17-26. DOI : 10.1145/192426.192435 (cf. p. 9).

[12] Michael Beham, Wolfgang Herzner, M. Eduard Gröller et Johannes Kehrer. « Cupid : cluster-based exploration of geometry generators with parallel coordinates and radial trees ». In : *Continuous scatterplots* 20.12 (2014), p. 1693-1702. ISSN : 1077-2626. DOI : 10.1109/tvcg.2014.2346626 (cf. p. 78).

[13] Michael Behrisch et al. « Quality Metrics for Information Visualization ». In : *Comput. Graph. Forum* 37.3 (2018), p. 625-662. DOI : 10.1111/cgf.13446 (cf. p. 6, 51, 52).

[14] Jacques Bertin. *La graphique et le traitement graphique de línformation*. Flammarion, 1977. ISBN : 2-08-211112-1 (cf. p. 1).

[15] Jacques Bertin. *Sémiologie graphique : les diagrammes-les réseaux-les cartes*. Gauthier-Villars Mouton & Cie, 1973 (cf. p. 6, 7, 45).

[16] Enrico Bertini, Alessio Di Girolamo et Giuseppe Santucci. « See What You Know : Analyzing Data Distribution to Improve Density Map Visualization ». In : *EuroVis07 : Joint Eurographics - IEEE VGTC Symposium on Visualization, Norrköping, Sweden, 23-25 May 2007*. 2007, p. 163-170. DOI : 10.2312/VisSym/EuroVis07/163-170 (cf. p. 45, 49, 50, 55).

[17] Enrico Bertini et Giuseppe Santucci. « Give chance a chance- modeling density to enhance scatter plot quality through random data sampling ». In : *Information Visualization* 5.2 (2006), p. 95-110. DOI : 10.1057/palgrave.ivs.9500122 (cf. p. 18, 48).

[18] Enrico Bertini et Giuseppe Santucci. « Quality Metrics for 2D Scatterplot Graphics : Automatically Reducing Visual Clutter ». In : *4th International Symposium, SG 2004, Banff, Canada, May 23-25, 2004, Proceedings*. T. 3031. 2004, p. 77-89. DOI : 10.1007/978-3-540-24678-7_8 (cf. p. 51).

[19] Staffan Björk, Lars Erik Holmquist et Johan Redström. « A Framework for Focus+Context Visualization ». In : *IEEE Symposium on Information Visualization 1999 (INFOVIS'99), San Francisco, California, USA, October 24-29, 1999*. 1999, p. 53-56. DOI : 10.1109/INFVIS.1999.801857 (cf. p. 9).

[20] Renaud Blanch et Eric Lecolinet. « Browsing Zoomable Treemaps : Structure-Aware Multi-Scale Navigation Techniques ». In : *IEEE Trans. Vis. Comput. Graph.* 13.6 (2007), p. 1248-1253. ISSN : om. DOI : 10.1109/TVCG.2007.70540 (cf. p. 16).

[21] Kenneth R. Boff, Lloyd Kaufman et James P. Thomas. *Handbook of Perception and Human Performance, Volumes 1 & 2*. Wiley, 1986. ISBN : 9780471829560 (cf. p. 16).

[22] S. Borzsony, D. Kossmann et K. Stocker. « The skyline operator ». In : *Proceedings 17th international conference on data engineering*. IEEE. 2001, p. 421-430. DOI : 10.1109/ICDE.2001.914855 (cf. p. 87).

[23] Reto Bürgin et Gilbert Ritschard. « A Decorated Parallel Coordinate Plot for Categorical Longitudinal Data ». In : *The American Statistician* 68.2 (2014), p. 98-103. DOI : 10.1080/00031305.2014.887591 (cf. p. 18).

[24] Thorsten Büring, Jens Gerken et Harald Reiterer. « User Interaction with Scatterplots on Small Screens - A Comparative Evaluation of Geometric-Semantic Zoom and Fisheye Distortion ». In : *IEEE Trans. Vis. Comput. Graph.* 12.5 (2006), p. 829-836. DOI : 10.1109/TVCG.2006.187 (cf. p. 18).

[25] K. Selçuk Candan, Luigi Di Caro et Maria Luisa Sapino. « PhC : Multiresolution Visualization and Exploration of Text Corpora with Parallel Hierarchical Coordinates ». In : *ACM TIST* 3.2 (2012), 22 :1-22 :36. DOI : 10.1145/2089094.2089098 (cf. p. 77, 90, 92).

[26] Stuart K. Card, Jock D. Mackinlay et Ben Shneiderman. *Readings in information visualization : using vision to think*. Morgan kaufmann publishers, 1999. ISBN : 978-1-55860-533-6 (cf. p. 1, 7, 11).

[27] Stuart K. Card, George G. Robertson et Jock D. Mackinlay. « The information visualizer, an information workspace ». In : *Conference on Human Factors in Computing Systems, CHI 1991, New Orleans, LA, USA, April 27 - May 2, 1991, Proceedings*. 1991, p. 181-186. DOI : 10.1145/108844.108874 (cf. p. 14, 15, 78).

[28] M. Sheelagh T. Carpendale, David J. Cowperthwaite et F. David Fracchia. « 3-Dimensional Pliable Surfaces : For the Effective Presentation of Visual Information ». In : *Proceedings of the 8th Annual ACM Symposium on User Interface Software and Technology, UIST 1995, Pittsburgh, PA, USA, November 14-17, 1995*. 1995, p. 217-226. DOI : 10.1145/215585.215978 (cf. p. 25).

[29] M. Sheelagh T. Carpendale, David J. Cowperthwaite, Margaret-Anne D. Storey et F. David Fracchia. *Exploring distinct aspects of the distortion viewing paradigm*. Rapp. tech. 97-08. School of computing science, Simon Fraser University, Burnaby, 1997 (cf. p. 25).

[30] M. Sheelagh T. Carpendale et Catherine Montagnese. « A framework for unifying presentation space ». In : *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, UIST 2001, Disney's BoardWalk Inn Resort, Walt Disney World, Orlando, Florida, USA, November 11-14, 2001*. 2001, p. 61-70. DOI : 10.1145/502348.502358 (cf. p. 25).

[31] Daniel B. Carr, Richard J. Littlefield et Wesley L. Nicholson. « Scatterplot Matrix Techniques for Large N ». In : *Proceedings of the Seventeenth Symposium on the Interface of Computer Sciences and Statistics on Computer Science and Statistics*. 1986, p. 297-306. ISBN : 0-444-70018-8. DOI : 10.1080/01621459.1987.10478445 (cf. p. 14, 18, 49).

[32] Chunlei Chang, Tim Dwyer et Kim Marriott. « An Evaluation of Perceptually Complementary Views for Multivariate Data ». In : *IEEE Pacific Visualization Symposium, PacificVis 2018, Kobe, Japan, April 10-13, 2018*. 2018, p. 195-204. DOI : 10.1109/PacificVis.2018.00033 (cf. p. 7).

[33] Haidong Chen et al. « Visual Abstraction and Exploration of Multi-class Scatterplots ». In : *IEEE Trans. Vis. Comput. Graph.* 20.12 (2014), p. 1683-1692. DOI : 10.1109/TVCG.2014.2346594 (cf. p. 18-20, 48).

[34] Jerry Chen, Ilmi Yoon et Wes Bethel. « Interactive, Internet Delivery of Visualization via Structured Prerendered Multiresolution Imagery ». In : *IEEE Trans. Vis. Comput. Graph.* 14.2 (2008), p. 302-312. DOI : 10.1109/TVCG.2007.70428 (cf. p. 46).

[35] Min Chen et Amos Golan. « What May Visualization Processes Optimize ? » In : *IEEE Trans. Vis. Comput. Graph.* 22.12 (2016), p. 2619-2632. DOI : 10.1109/TVCG.2015.2513410 (cf. p. 5).

[36] Min Chen et Heike Jänicke. « An Information-theoretic Framework for Visualization ». In : *IEEE Trans. Vis. Comput. Graph.* 16.6 (2010), p. 1206-1215. DOI : 10.1109/TVCG.2010.132 (cf. p. 52).

[37] Ed Huai-hsin Chi. « A Taxonomy of Visualization Techniques Using the Data State Reference Model ». In : *IEEE Symposium on Information Visualization 2000 (INFOVIS'00), Salt Lake City, Utah, USA, October 9-10, 2000*. 2000, p. 69-75. DOI : 10.1109/INFVIS.2000.885092 (cf. p. 11).

[38] William C. Cleveland et Marylyn E. Mcgill. *Dynamic graphics for statistics*. 1st. CRC Press, Inc., 1988. ISBN : 053409144X (cf. p. 6).

[39] Andy Cockburn, Amy Karlson et Benjamin B. Bederson. « A review of overview+detail, zooming, and focus+context interfaces ». In : *ACM computing surveys* 41.1 (2008), p. 1-31. ISSN : 0360-0300. DOI : 10.1145/1456650.1456652 (cf. p. 9).

[40] Christopher Collins et Sheelagh Carpendale. « VisLink : Revealing Relationships Amongst Visualizations ». In : *IEEE Trans. Vis. Comput. Graph.* 13.6 (2007), p. 1192-1199. DOI : 10.1109/TVCG.2007.70521 (cf. p. 28).

[41] Charles E. Cook, Ewan Birney, Guy Cochrane, Robert D. Finn, Rolf Apweiler et Mary Todd Bergman. « The european bioinformatics institute in 2016 : data growth and integration ». In : *Nucleic acids research* 44.D1 (2015), p. D20-d26. ISSN : 0305-1048. DOI : 10.1093/nar/gkv1352 (cf. p. 1).

[42] Joseph A. Cottam, Andrew Lumsdaine et Peter Wang. « Abstract rendering : out-of-core rendering for information visualization ». In : *Visualization and Data Analysis 2014, San Francisco, CA, USA, February 3-5, 2014*. T. 9017. 2014, 90170K. DOI : 10.1117/12.2041200 (cf. p. 15).

[43] Joseph A. Cottam, Andrew Lumsdaine et Peter Wang. « Overplotting : Unified solutions under Abstract Rendering ». In : *Proceedings of the 2013 IEEE International Conference on Big Data, 6-9 October 2013, Santa Clara, CA, USA*. 2013, p. 9-16. DOI : 10.1109/BigData.2013.6691712 (cf. p. 13).

[44] Qingguang Cui, Matthew Ward, Elke Rundensteiner et Jing Yang. « Measuring data abstraction quality in multiresolution visualizations ». In : *Continuous scatterplots* 12.5 (2006), p. 709-716. ISSN : 1077-2626. DOI : 10.1109/TVCG.2006.161 (cf. p. 52).

[45] Jeffrey Dean et Sanjay Ghemawat. « MapReduce : simplified data processing on large clusters ». In : *Commun. ACM* 51.1 (2008), p. 107-113. DOI : 10.1145/1327452.1327492 (cf. p. 16).

[46] Helmut Doleisch et Helwig Hauser. « Smooth Brushing for Focus+Context Visualization of Simulation Data in 3D ». In : *The 10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2002, WSCG 2002, University of West Bohemia, Campus Bory, Plzen-Bory, Czech Republic, February 4-8, 2002*. 2002, p. 147-154. URL : http://wscg.zcu.cz/wscg2002/Papers%5C%5C_2002/E71.pdf (cf. p. 10, 26, 28).

[47] Punit R. Doshi, Geraldine E. Rosario, Elke A. Rundensteiner et Matthew O. Ward. « A Strategy Selection Framework for Adaptive Prefetching in Data Visualization ». In : *Proceedings of the 15th International Conference on Scientific and Statistical Database Management (SSDBM)*. 2003, p. 107-116. DOI : 10.1109/SSDM.2003.1214972 (cf. p. 20).

[48] Dheeru Dua et Casey Graff. *UCI machine learning repository*. 2017. URL : http://archive.ics.uci.edu/ml (cf. p. 99).

[49] Jonathan Dubois, Amine Ghozlane, Patricia Thébault, Isabelle Dutour et Romain Bourqui. « Genome-wide detection of sRNA targets with rNAV ». In : *IEEE Symposium on Biological Data Visualization, BioVis 2013, Atlanta, GA, USA, October 13-14, 2013*. 2013, p. 81-88. DOI : 10.1109/BioVis.2013.6664350 (cf. p. 26, 28).

[50] Ahmed Eldawy, Mohamed F. Mokbel et Christopher Jonathan. « HadoopViz : A MapReduce framework for extensible visualization of big spatial data ». In : *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*. 2016, p. 601-612. DOI : 10.1109/ICDE.2016.7498274 (cf. p. 20).

[51] Geoffrey P. Ellis, Enrico Bertini et Alan J. Dix. « The sampling lens : making sense of saturated visualisations ». In : *Extended Abstracts Proceedings of the 2005 Conference on Human Factors in Computing Systems, CHI 2005, Portland, Oregon, USA, April 2-7, 2005*. 2005, p. 1351-1354. DOI : 10.1145/1056808.1056914 (cf. p. 18, 76).

[52] Geoffrey P. Ellis et Alan J. Dix. « A Taxonomy of Clutter Reduction for Information Visualisation ». In : *IEEE Trans. Vis. Comput. Graph.* 13.6 (2007), p. 1216-1223. DOI : 10.1109/TVCG.2007.70535 (cf. p. 13, 17, 18, 45).

[53] Geoffrey P. Ellis et Alan J. Dix. « The plot, the clutter, the sampling and its lens : occlusion measures for automatic clutter reduction ». In : *Proceedings of the working conference on Advanced visual interfaces, AVI 2006, Venezia, Italy, May 23-26, 2006*. 2006, p. 266-269. DOI : 10.1145/1133265.1133318 (cf. p. 18).

[54] Geoffrey Ellis et Alan Dix. « Enabling automatic clutter reduction in parallel coordinate plots ». In : *Continuous scatterplots* 12.5 (2006), p. 717-724. ISSN : 1077-2626. DOI : 10.1109/TVCG.2006.138 (cf. p. 18).

[55] Niklas Elmqvist et Jean-Daniel Fekete. « Hierarchical Aggregation for Information Visualization : Overview, Techniques, and Design Guidelines ». In : *IEEE Trans. Vis. Comput. Graph.* 16.3 (2010), p. 439-454. DOI : 10.1109/TVCG.2009.84 (cf. p. 8-10, 13, 19, 20).

[56] Niklas Elmqvist, Nathalie Henry, Yann Riche et Jean-Daniel Fekete. « Melange : space folding for multi-focus interaction ». In : *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008*. 2008, p. 1333-1342. DOI : 10.1145/1357054.1357263 (cf. p. 25, 28).

[57] Elena Fanea, Sheelagh Carpendale et Tobias Isenberg. « An Interactive 3D Integration of Parallel Coordinates and Star Glyphs ». In : *IEEE Symposium on Information Visualization (InfoVis 2005), 23-25 October 2005, Minneapolis, MN, USA*. 2005, p. 149-156. DOI : 10.1109/INFVIS.2005.1532141 (cf. p. 17, 18, 24, 25).

[58] Jean-Daniel Fekete et Catherine Plaisant. « Interactive Information Visualization of a Million Items ». In : *2002 IEEE Symposium on Information Visualization (InfoVis 2002), 27 October - 1 November 2002, Boston, MA, USA*. 2002, p. 117-124. DOI : 10.1109/INFVIS.2002.1173156 (cf. p. 16).

[59] Jean-Daniel Fekete, Jarke J. van Wijk, John T. Stasko et Chris North. « The Value of Information Visualization ». In : *Information Visualization - Human-Centered Issues and Perspectives*. T. 4950. 2008, p. 1-18. DOI : 10.1007/978-3-540-70956-5_1 (cf. p. 1).

[60] Danyel Fisher. « Big data exploration requires collaboration between visualization and data infrastructures ». In : *Proceedings of the workshop on human-in-the-loop data analytics - hilda Í6*. 2016, p. 1-5. ISBN : 978-1-4503-4207-0. DOI : 10.1145/2939502.2939518 (cf. p. 13).

[61] Steffen Frey, Filip Sadlo et Thomas Ertl. « Balanced sampling and compression for remote visualization ». In : *SIGGRAPH Asia 2015 Visualization in High Performance Computing, Kobe, Japan, November 2-6, 2015*. 2015, 1 :1-1 :4. DOI : 10.1145/2818517.2818529 (cf. p. 21).

[62] Ying-Huey Fua, Matthew O. Ward et Elke A. Rundensteiner. « Hierarchical Parallel Coordinates for Exploration of Large Datasets ». In : *IEEE Visualization 1999, 24-29 October 1999, San Francisco, CA, USA, Proceedings*. 1999, p. 43-50. DOI : 10.1109/VISUAL.1999.809866 (cf. p. 10, 18-20, 77, 78, 85, 91, 92).

[63] George W. Furnas. « Generalized Fisheye Views ». In : *SIGCHI Bull.* 17.4 (1986), p. 16-23. ISSN : 0736-6906. DOI : 10.1145/22339.22342 (cf. p. 9, 25).

[64] Emden R. Gansner, Yehuda Koren et Stephen C. North. « Topological Fisheye Views for Visualizing Large Graphs ». In : *IEEE Trans. Vis. Comput. Graph.* 11.4 (2005), p. 457-468. DOI : 10.1109/TVCG.2005.66 (cf. p. 40).

[65] Zhao Geng, Zhenmin Peng, Robert S. Laramee, Jonathan C. Roberts et Rick Walker. « Angular Histograms : Frequency-Based Visualizations for Large, High Dimensional Data ». In : *IEEE Trans. Vis. Comput. Graph.* 17.12 (2011), p. 2572-2580. DOI : 10.1109/TVCG.2011.166 (cf. p. 77).

[66] Parke Godfrey, Jarek Gryz et Piotr Lasek. « Interactive Visualization of Large Data Sets ». In : *IEEE Trans. Knowl. Data Eng.* 28.8 (2016), p. 2142-2157. DOI : 10.1109/TKDE.2016.2557324 (cf. p. 2, 13, 19, 46).

[67] Martin Graham et Jessie B. Kennedy. « Combining Linking & Focusing Techniques for a Multiple Hierarchy Visualisation ». In : *International Conference on Information Visualisation, IV 2001, London, England, UK, July 25-27, 2001*. 2001, p. 425-434. DOI : 10.1109/IV.2001.942092 (cf. p. 27, 28, 41).

[68] Martin Graham et Jessie B. Kennedy. « Using Curves to Enhance Parallel Coordinate Visualisations ». In : *Seventh International Conference on Information Visualization, IV 2003, 16-18 July 2003, London, UK*. 2003, p. 10-16. DOI : 10.1109/IV.2003.1217950 (cf. p. 18, 76).

[69] Kyle Wm. Hall, Charles Perin, Peter G. Kusalik, Carl Gutwin et Sheelagh Carpendale. « Formalizing Emphasis in Information Visualization ». In : *Comput. Graph. Forum* 35.3 (2016), p. 717-737. DOI : 10.1111/cgf.12936 (cf. p. 9, 23, 25, 26, 29, 39).

[70] Helwig Hauser. « Generalizing focus+context visualization ». In : *Scientific visualization : the visual extraction of knowledge from data*. 2006, p. 305-327. ISBN : 978-3-540-26066-0. DOI : 10.1007/3-540-30790-7_18 (cf. p. 25, 28).

[71] Helwig Hauser, Florian Ledermann et Helmut Doleisch. « Angular brushing of extended parallel coordinates ». In : *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.* 2002, p. 127-130. DOI : 10.1109/infvis.2002.1173157 (cf. p. 9).

[72] Christopher G. Healey, Kellogg S. Booth et James T. Enns. « Visualizing Real-Time Multivariate Data Using Preattentive Processing ». In : *ACM Trans. Model. Comput. Simul.* 5.3 (1995), p. 190-221. DOI : 10.1145/217853.217855 (cf. p. 14).

[73] Christopher G. Healey et James T. Enns. « Attention and Visual Memory in Visualization and Computer Graphics ». In : *IEEE Trans. Vis. Comput. Graph.* 18.7 (2012), p. 1170-1188. DOI : 10.1109/TVCG.2011.127 (cf. p. 14).

[74] Jeffrey Heer et Michael Bostock. « Crowdsourcing graphical perception : using mechanical turk to assess visualization design ». In : *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*. 2010, p. 203-212. DOI : 10.1145/1753326.1753357 (cf. p. 6).

[75] Jeffrey Heer et Stuart K. Card. « DOITrees revisited : scalable, space-constrained visualization of hierarchical data ». In : *Proceedings of the working conference on Advanced visual interfaces, AVI 2004, Gallipoli, Italy, May 25-28, 2004*. 2004, p. 421-424. DOI : 10.1145/989863.989941 (cf. p. 8, 9).

[76] Jeffrey Heer et Ben Shneiderman. « Interactive Dynamics for Visual Analysis ». In : *Commun. ACM* 55.4 (2012), p. 45-54. ISSN : 0001-0782. DOI : 10.1145/2133806.2133821 (cf. p. 7, 8, 15).

[77] Julian Heinrich, John T. Stasko et Daniel Weiskopf. « The Parallel Coordinates Matrix ». In : *Eurographics Conference on Visualization, EuroVis 2012, Vienna, Austria, June 5-8, 2012*. 2012. DOI : 10.2312/PE/EuroVisShort/EuroVisShort2012/037-041 (cf. p. 75, 80, 81).

[78] Julian Heinrich et Daniel Weiskopf. « Continuous Parallel Coordinates ». In : *IEEE Trans. Vis. Comput. Graph.* 15.6 (2009), p. 1531-1538. DOI : 10.1109/TVCG.2009.131 (cf. p. 18, 19, 77).

[79] Julian Heinrich et Daniel Weiskopf. « State of the Art of Parallel Coordinates ». In : *Eurographics 2013 - State of the Art Reports, Girona, Spain, May 6-10, 2013*. 2013, p. 95-116. DOI : 10.2312/conf/EG2013/stars/095-116 (cf. p. 18, 19, 76).

[80] Martin Hilbert et Priscila López. « The World's Technological Capacity to Store, Communicate, and Compute Information ». In : *Science* 332.6025 (2011), p. 60-65. DOI : 10.1126/science.1200970 (cf. p. 1).

[81] Danny Holten et Jarke J. Van Wijk. « Evaluation of cluster identification performance for different pcp variants ». En. In : *Computer graphics forum* 29.3 (2010), p. 793-802. ISSN : 0167-7055. DOI : 10.1111/j.1467-8659.2009.01666.x (cf. p. 76).

[82] Alfred Inselberg et Bernard Dimsdale. « Parallel Coordinates : A Tool for Visualizing Multi-dimensional Geometry ». In : *Proceedings IEEE Visualization '90, San Francisco, California, USA, October 23-26, 1990*. 1990, p. 361-378. DOI : 10.1109/VISUAL.1990.146402 (cf. p. 6, 75).

[83] Željko Ivezić et al. « LSST : From Science Drivers to Reference Design and Anticipated Data Products ». In : *The Astrophysical Journal* 873.2 (2019), p. 111. DOI : 10.3847/1538-4357/ab042c (cf. p. 1).

[84] Halldor Janetzko, Ming C. Hao, Sebastian Mittelstädt, Umeshwar Dayal et Daniel A. Keim. « Enhancing Scatter Plots Using Ellipsoid Pixel Placement and Shading ». In : *46th Hawaii International Conference on System Sciences, HICSS 2013, Wailea, HI, USA, January 7-10, 2013*. 2013, p. 1522-1531. DOI : 10.1109/HICSS.2013.197 (cf. p. 70).

[85] Dean F. Jerding et John T. Stasko. « The Information Mural : A Technique for Displaying and Navigating Large Information Spaces ». In : *IEEE Trans. Vis. Comput. Graph.* 4.3 (1998), p. 257-271. DOI : 10.1109/2945.722299 (cf. p. 50).

[86] Jaemin Jo, Wonjae Kim, Seunghoon Yoo, Bo Hyoung Kim et Jinwook Seo. « SwiftTuna : Responsive and incremental visual exploration of large-scale multidimensional data ». In : *2017 IEEE Pacific Visualization Symposium, PacificVis 2017, Seoul, South Korea, April 18-21, 2017*. 2017, p. 131-140. DOI : 10.1109/PACIFICVIS.2017.8031587 (cf. p. 20, 21).

[87] Jaemin Jo, Frédéric Vernier, Pierre Dragicevic et Jean-Daniel Fekete. « A Declarative Rendering Model for Multiclass Density Maps ». In : *Continuous scatterplots* 25 (1 2019), p. 470-480. ISSN : 1077-2626. DOI : 10.1109/TVCG.2018.2865141 (cf. p. 49, 104).

[88] Jimmy Johansson et Matthew D. Cooper. « A Screen Space Quality Method for Data Abstraction ». In : *Comput. Graph. Forum* 27.3 (2008), p. 1039-1046. DOI : 10.1111/j.1467-8659.2008.01240.x (cf. p. 51, 52).

[89] Jimmy Johansson, Patric Ljung, Mikael Jern et Matthew D. Cooper. « Revealing Structure within Clustered Parallel Coordinates Displays ». In : *IEEE Symposium on Information Visualization (InfoVis 2005), 23-25 October 2005, Minneapolis, MN, USA*. 2005, p. 125-132. DOI : 10.1109/INFVIS.2005.1532138 (cf. p. 77, 78).

[90] Brian Johnson et Ben Shneiderman. « Tree maps : A Space-Filling Approach to the Visualization of Hierarchical Information Structures ». In : *Proceedings IEEE Visualization '91, San Diego, California, USA, October 22-25, 1991*. 1991, p. 284-291. DOI : 10.1109/VISUAL.1991.175815 (cf. p. 5, 6).

[91] Alan Keahey et Edward L. Robertson. « Nonlinear Magnification Fields ». In : *1997 IEEE Symposium on Information Visualization (InfoVis '97), October 18-25, 1997, Phoenix, AZ, USA*. 1997, p. 51-58. DOI : 10.1109/INFVIS.1997.636786 (cf. p. 25).

[92] Alan Keahey et Edward L. Robertson. « Techniques for non-linear magnification transformations ». In : *Proceedings of the IEEE Symposium on Information Visualization 1996, InfoVis '96, San Francisco, CA, USA, October 28-29, 1996*. 1996, p. 38-45. DOI : 10.1109/INFVIS.1996.559214 (cf. p. 25).

[93] Daniel A. Keim, Ming C. Hao, Umeshwar Dayal, Halldor Janetzko et Peter Bak. « Generalized scatter plots ». In : *Information visualization* 9.4 (2010), p. 301-311. ISSN : 1473-8716. DOI : 10.1057/ivs.2009.34 (cf. p. 18, 25).

[94] Daniel A. Keim et Annemarie Herrmann. « The Gridfit algorithm : an efficient and effective approach to visualizing large amounts of spatial data ». In : *Visualization '98, Proceedings, October 18-23, 1998, Research Triangle Park, North Carolina, USA*. 1998, p. 181-188. DOI : 10.1109/VISUAL.1998.745301 (cf. p. 18).

[95] Gordon L. Kindlmann et Carlos Eduardo Scheidegger. « An Algebraic Process for Visualization Design ». In : *IEEE Trans. Vis. Comput. Graph.* 20.12 (2014), p. 2181-2190. DOI : 10.1109/TVCG.2014.2346325 (cf. p. 6).

[96] Robert Kosara, Fabian Bendix et Helwig Hauser. « Parallel Sets : Interactive Exploration and Visual Analysis of Categorical Data ». In : *IEEE Trans. Vis. Comput. Graph.* 12.4 (2006), p. 558-568. DOI : 10.1109/TVCG.2006.76 (cf. p. 77, 82).

[97] Philipp Koytek, Charles Perin, Jo Vermeulen, Elisabeth André et Sheelagh Carpendale. « MyBrush : Brushing and Linking with Personal Agency ». In : *IEEE Trans. Vis. Comput. Graph.* 24.1 (2018), p. 605-615. DOI : 10.1109/TVCG.2017.2743859 (cf. p. 10, 25).

[98] Antoine Lambert. « Visualisation interactive de graphes : élaboration et optimisation dálgorithmes à coûts computationnels élevés ». Thèse de doctorat dirigée par David Auber, Informatique, Bordeaux 2017. Thèse de doct. 2012. URL : http://theses.fr/2012BOR14664 (cf. p. 25).

[99] John Lamping et Ramana Rao. « The Hyperbolic Browser : A Focus + Context Technique for Visualizing Large Hierarchies ». In : *J. Vis. Lang. Comput.* 7.1 (1996), p. 33-55. DOI : 10.1006/jvlc.1996.0003 (cf. p. 25).

[100] Cicero Augusto de Lara Pahins, Sean A. Stephens, Carlos Scheidegger et João Luiz Dihl Comba. « Hashedcubes : Simple, Low Memory, Real-Time Visual Exploration of Big Data ». In : *IEEE Trans. Vis. Comput. Graph.* 23.1 (2017), p. 671-680. DOI : 10.1109/TVCG.2016.2598624 (cf. p. 20, 49).

[101] Ying K. Leung et Mark D. Apperley. « A Review and Taxonomy of Distortion-Oriented Presentation Techniques ». In : *ACM Trans. Comput.-Hum. Interact.* 1.2 (1994), p. 126-160. DOI : 10.1145/180171.180173 (cf. p. 25, 30).

[102] Chenhui Li, George Baciu et Yu Han. « Interactive visualization of high density streaming points with heat-map ». In : *International Conference on Smart Computing, SMARTCOMP 2014, Hong Kong, China, November 3-5, 2014*. 2014, p. 145-149. DOI : 10.1109/SMARTCOMP.2014.7043852 (cf. p. 19).

[103] Mingzhao Li, Farhana Murtaza Choudhury, Zhifeng Bao, Hanan Samet et Timos Sellis. « ConcaveCubes : Supporting Cluster-based Geographical Visualization in Large Data Scale ». In : *Comput. Graph. Forum* 37.3 (2018), p. 217-228. DOI : 10.1111/cgf.13414 (cf. p. 49, 50, 70).

[104] Mats Lind, Jimmy Johansson et Matthew D. Cooper. « Many-to-Many Relational Parallel Coordinates Displays ». In : *13th International Conference on Information Visualisation, IV 2009, 15-17 July 2009, Barcelona, Spain*. 2009, p. 25-31. DOI : 10.1109/IV.2009.43 (cf. p. 75, 80, 81).

[105] Lauro Didier Lins, James T. Klosowski et Carlos Eduardo Scheidegger. « Nanocubes for Real-Time Exploration of Spatiotemporal Datasets ». In : *IEEE Trans. Vis. Comput. Graph.* 19.12 (2013), p. 2456-2465. DOI : 10.1109/TVCG.2013.179 (cf. p. 20, 49, 50).

[106] Zhicheng Liu et Jeffrey Heer. « The effects of interactive latency on exploratory visual analysis ». In : *IEEE Trans. Vis. Comput. Graph.* 20.12 (2014), p. 2122-2131. DOI : 10.1109/TVCG.2014.2346452 (cf. p. 2, 13, 15).

[107] Zhicheng Liu, Biye Jiang et Jeffrey Heer. « *imMens* : Real-time Visual Querying of Big Data ». In : *Comput. Graph. Forum* 32.3 (2013), p. 421-430. DOI : 10.1111/cgf.12129 (cf. p. 2, 14, 20, 47, 49, 70).

[108] Laurens Van Der Maaten et Geoffrey Hinton. « Visualizing data using t-SNE ». In : *Journal of Machine Learning Research* 9.Nov (2008), p. 2579-2605 (cf. p. 6).

[109] Jock D. Mackinlay. « Automating the Design of Graphical Presentations of Relational Information ». In : *ACM Trans. Graph.* 5.2 (1986), p. 110-141. DOI : 10.1145/22949.22950 (cf. p. 6).

[110] Nathan Marz et James Warren. *Big data : principles and best practices of scalable realtime data systems*. 1re éd. Manning publications co., 2015. ISBN : 1617290343 (cf. p. 1).

[111] Kresimir Matkovic, Wolfgang Freiler, Denis Gracanin et Helwig Hauser. « ComVis : A Coordinated Multiple Views System for Prototyping New Visualization Technology ». In : *12th International Conference on Information Visualisation, IV 2008, 8-11 July 2008, London, UK*. 2008, p. 215-220. DOI : 10.1109/IV.2008.87 (cf. p. 28).

[112] Viktor Mayer-Schönberger et Kenneth Cukier. *Big Data : A Revolution That Will Transform How We Live, Work, and Think*. Eamon Dolan/Houghton Mifflin Harcourt, 2013. ISBN : 978-0-544-00269-2 (cf. p. 1).

[113] Andrew Mccallum, Kamal Nigam et Lyle H. Ungar. « Efficient clustering of high-dimensional data sets with application to reference matching ». In : *Kdd*. 2000, p. 169-178. DOI : 10.1145/347090.347123 (cf. p. 58, 97).

[114] Kevin T. McDonnell et Klaus Mueller. « Illustrative Parallel Coordinates ». In : *Comput. Graph. Forum* 27.3 (2008), p. 1031-1038. DOI : 10.1111/j.1467-8659.2008.01239.x (cf. p. 18, 19, 76, 77).

[115] Luana Micallef, Gregorio Palmas, Antti Oulasvirta et Tino Weinkauf. « Towards Perceptual Optimization of the Visual Design of Scatterplots ». In : *IEEE Trans. Vis. Comput. Graph.* 23.6 (2017), p. 1588-1599. DOI : 10.1109/TVCG.2017.2674978 (cf. p. 6, 18, 51, 52).

[116] Charles-Joseph Minard. *Des tableaux graphiques et des cartes figuratives*. 1862. URL : https://gallica.bnf.fr/ark:/12148/btv1b53074870p (cf. p. 1, 2).

[117] Kenneth Moreland. « A Survey of Visualization Pipelines ». In : *IEEE Trans. Vis. Comput. Graph.* 19.3 (2013), p. 367-378. DOI : 10.1109/TVCG.2012.133 (cf. p. 15).

[118] Dominik Moritz, Jeffrey Heer et Bill Howe. « Dynamic client-server optimization for scalable interactive visualization on the web ». In : *The 1st workshop on data systems for interactive analysis (DSIA) at VIS 2015*. 2015, p. 5. URL : http://www.interactive-analysis.org/papers/2015/moritz.pdf (cf. p. 12).

[119] Dominik Moritz, Bill Howe et Jeffrey Heer. « Falcon : Balancing Interactive Latency and Resolution Sensitivity for Scalable Linked Visualizations ». In : *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*. 2019, p. 694. DOI : 10.1145/3290605.3300924 (cf. p. 20).

[120] Tomer Moscovich, Fanny Chevalier, Nathalie Henry, Emmanuel Pietriga et Jean-Daniel Fekete. « Topology-aware navigation in large networks ». In : *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*. 2009, p. 2319-2328. DOI : 10.1145/1518701.1519056 (cf. p. 26).

[121] Tamara Munzner. *Visualization analysis and design*. CRC Press, 2014. ISBN : 9781466508910 (cf. p. 5, 6, 11, 14, 23).

[122] Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang et Yunhong Zhou. « TreeJuxtaposer : scalable tree comparison using Focus+Context with guaranteed visibility ». In : *ACM Trans. Graph.* 22.3 (2003), p. 453-462. DOI : 10.1145/882262.882291 (cf. p. 27, 28, 37).

[123] Petra Neumann et Sheelagh Carpendale. *Taxonomy for discrete lenses*. Rapp. tech. 2003-734-37. Department of computer science, University of Calgary, 2003 (cf. p. 25).

[124] Hoa Nguyen et Paul Rosen. « DSPCP : A Data Scalable Approach for Identifying Relationships in Parallel Coordinates ». In : *IEEE Trans. Vis. Comput. Graph.* 24.3 (2018), p. 1301-1315. DOI : 10.1109/TVCG.2017.2661309 (cf. p. 77, 78, 90).

[125] Lianqiang Niu, Shidong Yu et Shengnan Zhang. « Study on Enhancing Visuality of Parallel Coordinates ». In : 2007, p. 530-530. DOI : 10.1109/ICICIC.2007.547 (cf. p. 25, 76).

[126] Chris North et Ben Shneiderman. « Snap-together visualization : can users construct and operate coordinated visualizations? » In : *Int. J. Hum.-Comput. Stud.* 53.5 (2000), p. 715-739. DOI : 10.1006/ijhc.2000.0418 (cf. p. 25).

[127] Matej Novotný et Helwig Hauser. « Outlier-Preserving Focus+Context Visualization in Parallel Coordinates ». In : *IEEE Trans. Vis. Comput. Graph.* 12.5 (2006), p. 893-900. DOI : 10.1109/TVCG.2006.170 (cf. p. 9, 77, 90, 92).

[128] Lace M. K. Padilla, P. Samuel Quinan, Miriah D. Meyer et Sarah H. Creem-Regehr. « Evaluating the Impact of Binning 2D Scalar Fields ». In : *IEEE Trans. Vis. Comput. Graph.* 23.1 (2017), p. 431-440. DOI : 10.1109/TVCG.2016.2599106 (cf. p. 71).

[129] Gregorio Palmas, Myroslav Bachynskyi, Antti Oulasvirta, Hans-Peter Seidel et Tino Weinkauf. « An Edge-Bundling Layout for Interactive Parallel Coordinates ». In : *IEEE Pacific Visualization Symposium, PacificVis 2014, Yokohama, Japan, March 4-7, 2014*. 2014, p. 57-64. DOI : 10.1109/PacificVis.2014.40 (cf. p. 77, 82, 90, 91).

[130] Nilma Perera, Albert Goodman et Kathy Blashki. « Preattentive processing : using low-level vision psychology to encode information in visualisations ». In : *Proceedings of the 2007 ACM Symposium on Applied Computing (SAC), Seoul, Korea, March 11-15, 2007*. 2007, p. 1090-1091. DOI : 10.1145/1244002.1244240 (cf. p. 6).

[131] Alexandre Perrot. « La visualisation d'information à l'ère du Big Data : résoudre les problèmes de scalabilité par l'abstraction multi-échelle ». Thèse de doctorat dirigée par david auber, informatique bordeaux 2017. Thèse de doct. 2017. URL : http://www.theses.fr/2017BORD0775 (cf. p. 58).

[132] Alexandre Perrot et David Auber. « Cornac : Tackling Huge Graph Visualization with Big Data Infrastructure ». In : *IEEE Transactions on Big Data* PP (99 2018), p. 1-1. DOI : 10.1109/TBDATA.2018.2869165 (cf. p. 67).

[133] Alexandre Perrot, Romain Bourqui, Nicolas Hanusse, Frédéric Lalanne et David Auber. « Large interactive visualization of density functions on big data infrastructure ». In : *5th IEEE Symposium on Large Data Analysis and Visualization, LDAV 2015, Chicago, IL, USA, October 25-26, 2015*. 2015, p. 99-106. DOI : 10.1109/LDAV.2015.7348077 (cf. p. 12, 16, 19-21, 45, 47, 48, 50, 52).

[134] Harald Piringer, Christian Tominski, Philipp Muigg et Wolfgang Berger. « A Multi-Threading Architecture to Support Interactive Visual Exploration ». In : *IEEE Trans. Vis. Comput. Graph.* 15.6 (2009), p. 1113-1120. DOI : 10.1109/TVCG.2009.110 (cf. p. 11).

[135] Zening Qu et Jessica Hullman. « Keeping Multiple Views Consistent : Constraints, Validations, and Exceptions in Visualization Authoring ». In : *IEEE Trans. Vis. Comput. Graph.* 24.1 (2018), p. 468-477. DOI : 10.1109/TVCG.2017.2744198 (cf. p. 6, 24).

[136] Ramana Rao et Stuart K. Card. « The table lens : merging graphical and symbolic representations in an interactive focus+context visualization for tabular information ». In : *Conference on Human Factors in Computing Systems, CHI 1994, Boston, Massachusetts, USA, April 24-28, 1994, Conference Companion*. 1994, p. 222. DOI : 10.1145/259963.260391 (cf. p. 25).

[137] Jonathan C. Roberts. « Multiple view and multiform visualization ». In : (2000). DOI : 10.1117/12.378894 (cf. p. 27).

[138] Jonathan C. Roberts. « On encouraging multiple views for visualization ». In : 1998, p. 8-14. DOI : 10.1109/IV.1998.694193 (cf. p. 10).

[139] Jonathan C. Roberts. « State of the Art : Coordinated & Multiple Views in Exploratory Visualization ». In : *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*. 2007, p. 61-71. ISBN : 0-7695-2903-8. DOI : 10.1109/CMV.2007.20 (cf. p. 7, 10).

[140] Oliver Rübel et al. « High performance multivariate visual data exploration for extremely large data ». In : *Proceedings of the ACM/IEEE Conference on High Performance Computing, SC 2008, November 15-21, 2008, Austin, Texas, USA*. 2008, p. 51. DOI : 10.1109/SC.2008.5214436 (cf. p. 20, 71, 78).

[141] Selan R. dos Santos et Ken Brodlie. « Gaining understanding of multivariate and multidimensional data through visualization ». In : *Computers & Graphics* 28.3 (2004), p. 311-325. DOI : 10.1016/j.cag.2004.03.013 (cf. p. 11).

[142] Alper Sarikaya, Michael Gleicher et Danielle Albers Szafir. « Design Factors for Summary Visualization in Visual Analytics ». In : *Comput. Graph. Forum* 37.3 (2018), p. 145-156. DOI : 10.1111/cgf.13408 (cf. p. 61).

[143] Manojit Sarkar et Marc H. Brown. « Graphical Fisheye Views ». In : *Commun. ACM* 37.12 (1994), p. 73-84. DOI : 10.1145/198366.198384 (cf. p. 8).

[144] Manojit Sarkar, Scott S. Snibbe, Oren J. Tversky et Steven P. Reiss. « Stretching the Rubber Sheet : A Metaphor for Viewing Large Layouts on Small Screens ». In : *Proceedings of the Sixth ACM Symposium on User Interface Software and Technology, UIST 1993, Atlanta, GA, USA, November 3-5, 1993*. 1993, p. 81-91. DOI : 10.1145/168642.168650 (cf. p. 25, 28).

[145] Maximilian Scherr. *Multiple and coordinated views in information visualization*. Technical report Lmu-mi-20 10 -1, apr . 20 10. Ludwig maximilias university munich, 2008, p. 38-45 (cf. p. 7, 10).

[146] Hans-Jörg Schulz, Marco Angelini, Giuseppe Santucci et Heidrun Schumann. « An Enhanced Visualization Process Model for Incremental Visualization ». In : *IEEE Trans. Vis. Comput. Graph.* 22.7 (2016), p. 1830-1842. DOI : 10.1109/TVCG.2015.2462356 (cf. p. 21).

[147] David W. Scott. « A note on choice of bivariate histogram bin shape ». In : *Journal of Official Statistics* 4.1 (1988), p. 47-51 (cf. p. 49).

[148] Ben Shneiderman. « The Eyes Have It : A Task by Data Type Taxonomy for Information Visualizations ». In : *Proceedings of the 1996 IEEE Symposium on Visual Languages, Boulder, Colorado, USA, September 3-6, 1996*. 1996, p. 336-343. DOI : 10.1109/VL.1996.545307 (cf. p. 8).

[149] Daniel J. Simons. « Current Approaches to Change Blindness ». In : 7 (2000), p. 1-15. ISSN : 1350-6285. DOI : 10.1080/135062800394658 (cf. p. 14).

[150] Stephen Smart et Danielle Albers Szafir. « Measuring the Separability of Shape, Size, and Color in Scatterplots ». In : *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*. 2019, p. 669. DOI : 10.1145/3290605.3300899 (cf. p. 23, 29).

[151] Markus Steinberger, Manuela Waldner, Marc Streit, Alexander Lex et Dieter Schmalstieg. « Context-Preserving Visual Links ». In : *IEEE Trans. Vis. Comput. Graph.* 17.12 (2011), p. 2249-2258. DOI : 10.1109/TVCG.2011.183 (cf. p. 26, 28).

[152] S. S. Stevens. « On the Theory of Scales of Measurement ». In : *Science* 103.2684 (1946), p. 677-680. DOI : 10.1126/science.103.2684.677 (cf. p. 6).

[153] Charles D. Stolper, Adam Perer et David Gotz. « Progressive Visual Analytics : User-Driven Visual Exploration of In-Progress Analytics ». In : *IEEE Trans. Vis. Comput. Graph.* 20.12 (2014), p. 1653-1662. DOI : 10.1109/TVCG.2014.2346574 (cf. p. 21).

[154] Danielle Albers Szafir. « Modeling Color Difference for Visualization Design ». In : *IEEE Trans. Vis. Comput. Graph.* 24.1 (2018), p. 392-401. DOI : 10.1109/TVCG.2017.2744359 (cf. p. 6).

[155] Anne Treisman. « Preattentive processing in vision ». In : *Computer vision, graphics, and image processing* 31.2 (1985), p. 156-177. ISSN : 0734-189X. DOI : 10.1016/S0734-189X(85)80004-9 (cf. p. 1).

[156] Marjan Trutschl, Georges G. Grinstein et Urska Cvek. « Intelligently Resolving Point Occlusion ». In : *9th IEEE Symposium on Information Visualization (InfoVis 2003), 20-21 October 2003, Seattle, WA, USA*. 2003, p. 131-136. DOI : 10.1109/INFVIS.2003.1249018 (cf. p. 17, 18).

[157] Ying Tu et Han-Wei Shen. « Balloon Focus : a Seamless Multi-Focus+Context Method for Treemaps ». In : *IEEE Trans. Vis. Comput. Graph.* 14.6 (2008), p. 1157-1164. DOI : 10.1109/TVCG.2008.114 (cf. p. 25, 28).

[158] Edward R. Tufte. *The visual display of quantitative information*. 2nd ed. Graphics ress, 2001. ISBN : 0-9613921-4-2 (cf. p. 1).

[159] Jeffrey Scott Vitter. « External Memory Algorithms and Data Structures : Dealing with Massive Data ». In : *ACM Computing Surveys* 33.2 (2001), p. 209-271. ISSN : 0360-0300. DOI : 10/d3nmr7 (cf. p. 15).

[160] Huy T. Vo, João L. D. Comba, Berk Geveci et Cláudio T. Silva. « Streaming-Enabled Parallel Data Flow Framework in the Visualization ToolKit ». In : *Computing in Science Engineering* 13.5 (2011), p. 72-83. ISSN : 1521-9615. DOI : 10/bvdpds (cf. p. 15).

[161] Zana Vosough, Marius Hogräfer, Loïc Alain Royer, Rainer Groh et Hans-Jörg Schulz. « Parallel hierarchies : A visualization for cross-tabulating hierarchical categories ». In : *Computers & Graphics* 76 (2018), p. 1-17. DOI : 10.1016/j.cag.2018.07.009 (cf. p. 91, 92, 100).

[162] Junpeng Wang, Xiaotong Liu, Han-Wei Shen et Guang Lin. « Multi-Resolution Climate Ensemble Parameter Analysis with Nested Parallel Coordinates Plots ». In : *IEEE Trans. Vis. Comput. Graph.* 23.1 (2017), p. 81-90. DOI : 10.1109/TVCG.2016.2598830 (cf. p. 76).

[163] Zhou Wang et Alan C. Bovik. *Modern Image Quality Assessment*. Morgan & Claypool Publishers, 2006. DOI : 10.2200/S00010ED1V01Y200508IVM003 (cf. p. 52).

[164] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh et Eero P. Simoncelli. « Image quality assessment : from error visibility to structural similarity ». In : *IEEE Trans. Image Processing* 13.4 (2004), p. 600-612. DOI : 10.1109/TIP.2003.819861 (cf. p. 48, 52).

[165] Colin Ware. *Information visualization : perception for design*. Elsevier, 2012 (cf. p. 5, 6).

[166] Chris Weaver. « Building Highly-Coordinated Visualizations in Improvise ». In : *10th IEEE Symposium on Information Visualization (InfoVis 2004), 10-12 October 2004, Austin, TX, USA*. 2004, p. 159-166. DOI : 10.1109/INFVIS.2004.12 (cf. p. 24).

[167] Hadley Wickham. *Bin-summarise-smooth : a framework for visualising large data*. Rapp. tech. Had.co.nz, 2013 (cf. p. 18, 19, 49, 50).

[168] Jarke J. van Wijk. « Views on Visualization ». In : *IEEE Trans. Vis. Comput. Graph.* 12.4 (2006), p. 421-433. DOI : 10.1109/TVCG.2006.80 (cf. p. 6).

[169] Allison Woodruff, James A. Landay et Michael Stonebraker. « Constant Density Visualizations of Non-Uniform Distributions of Data ». In : *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, UIST 1998, San Francisco, CA, USA, November 1-4, 1998*. 1998, p. 19-28. DOI : 10.1145/288392.288397 (cf. p. 18).

[170] Yingcai Wu, Xiaotong Liu, Shixia Liu et Kwan-Liu Ma. « ViSizer : A Visualization Resizing Framework ». In : *IEEE Trans. Vis. Comput. Graph.* 19.2 (2013), p. 278-290. DOI : 10.1109/TVCG.2012.114 (cf. p. 41).

[171] Jing Yang, Matthew O. Ward et Elke A. Rundensteiner. « Interactive hierarchical displays : a general framework for visualization and exploration of large multivariate data sets ». In : *Computers & Graphics* 27.2 (2003), p. 265-283. DOI : 10.1016/S0097-8493(02)00283-2 (cf. p. 18, 49, 50, 70).

[172] Ji Soo Yi, Youn ah Kang, John T. Stasko et Julie A. Jacko. « Toward a Deeper Understanding of the Role of Interaction in Information Visualization ». In : *IEEE Trans. Vis. Comput. Graph.* 13.6 (2007), p. 1224-1231. DOI : 10.1109/TVCG.2007.70515 (cf. p. 8).

[173] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker et Ion Stoica. « Spark : Cluster Computing with Working Sets ». In : *2nd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud'10, Boston, MA, USA, June 22, 2010*. 2010. URL : https://www.usenix.org/conference/hotcloud-10/spark-cluster-computing-working-sets (cf. p. 16).

[174] Matei Zaharia et al. « Apache Spark : a unified engine for big data processing ». In : *Commun. ACM* 59.11 (2016), p. 56-65. DOI : 10.1145/2934664 (cf. p. 16).

[175] Xin Zhao, Wei Zeng, Xianfeng David Gu, Arie E. Kaufman, Wei Xu et Klaus Mueller. « Conformal Magnifier : A Focus+Context Technique with Local Shape Preservation ». In : *IEEE Trans. Vis. Comput. Graph.* 18.11 (2012), p. 1928-1941. DOI : 10.1109/TVCG.2012.70 (cf. p. 41).

[176] Yan Zheng, Jeffrey Jestes, Jeff M. Phillips et Feifei Li. « Quality and efficiency for kernel density estimates in large data ». In : *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*. 2013, p. 433-444. DOI : 10.1145/2463676.2465319 (cf. p. 48).

[177] Yan Zheng, Yi Ou, Alexander Lex et Jeff M. Phillips. « Visualization of Big Spatial Data using Coresets for Kernel Density Estimates ». In : *IEEE Transactions on Big Data, early access* (2019). DOI : 10.1109/TBDATA.2019.2913655 (cf. p. 48).

[178] Hong Zhou, Xiaoru Yuan, Huamin Qu, Weiwei Cui et Baoquan Chen. « Visual Clustering in Parallel Coordinates ». In : *Comput. Graph. Forum* 27.3 (2008), p. 1047-1054. DOI : 10.1111/j.1467-8659.2008.01241.x (cf. p. 76).

[179] Caroline Ziemkiewicz et Robert Kosara. « Embedding information visualization within visual representation ». In : *Advances in information and intelligent systems*. T. 251. 2009, p. 307-326. ISBN : 978-3-642-04140-2. DOI : 10.1007/978-3-642-04141-9_15 (cf. p. 6).

[180] Thomas Zinner, Oliver Hohlfeld, Osama Abboud et Tobias Hossfeld. « Impact of frame rate and resolution on objective QoE metrics ». In : 2010, p. 29-34. DOI : 10.1109/QOMEX.2010.5518277 (cf. p. 52).

Liste des publications

Articles en revues internationales

[1] Gaëlle Richer, Joris Sansen, Frédéric Lalanne, David Auber et Romain Bourqui. « HiePaCo : Interactive exploration in hierarchical parallel coordinates with a graph-based model for large multidimensional data ». In : *Big Data Research* (2019). doi : 10.1016/j.bdr.2019.07.001.

[2] Joris Sansen, Gaëlle Richer, Timothée Jourde, Frédéric Lalanne, David Auber et Romain Bourqui. « Visual Exploration of Large Multidimensional Data Using Parallel Coordinates on Big Data Infrastructure ». In : *Informatics* 4.3 (2017), p. 21. doi : 10.3390/informatics4030021.

Articles dans les actes de conférences internationales

[3] Gaëlle Richer, Romain Bourqui et David Auber. « CorFish : Coordinating Emphasis Across Multiple Views Using Spatial Distortion ». In : *2019 IEEE Pacific Visualization Symposium (PacificVis)*. 2019, p. 1-10. doi : 10.1109/PacificVis.2019.00009.

Articles dans les actes de colloques internationaux

[4] Gaëlle Richer, Joris Sansen, Frédéric Lalanne, David Auber et Romain Bourqui. « Enabling Hierarchical Exploration for Large-Scale Multidimensional Data with Abstract Parallel Coordinates ». In : *Proceedings of the Workshops of the EDBT/ICDT 2018 Joint Conference (EDBT/ICDT 2018), Vienne, Autriche, 26 mars 2018*. 2018, p. 76-83. url : http://ceur-ws.org/Vol-2083/paper-12.pdf.

Liste des figures

Liste des tableaux

Glossaire

binning partitionnement par pavage régulier de l'espace (avec des rectangles ou hexagones) qui sert en général à partitionner des données pour les représenter de manière agrégée . 3, 49, 77, 90, 103

jittering déplacement des entités visuelles autour de leur position d'origine ayant pour objectif d'éviter leur superposition et d'augmenter leur visibilité. 17, 18, 45

treemap représentation visuelle de données hiérarchique (arbre) utilisant la propriété d'inclusion géomérique (entre des rectangles ou cercles généralement) pour représenter le lien de parenté entre deux entités . 5, 6, 38, 39, 40, 41

brossage et lien (*brushing & linking*, aussi *linked highlighting*) technique d'interaction permettant à l'utilisateur de définir une sélection sur une partie de la visualisation provoquant la mise en surbrillance de toutes les instances liées à cette sélection. 7, 9, 10, 14, 15, 20, 23, 24, 26, 40, 42, 75, 104

CDD carte de densité discrète. 55, 61, 71

CPU processeur (*central processing unit*). 15

DOI degré d'intérêt (*degree of interest*). 9, 10, 25

encombrement visuel (*visual clutter*) accumulation excessive locale ou globale d'éléments graphiques dont la densité voire la superposition peut nuire à la compréhension de la visualisation voire dissimuler des éléments jusqu'à entraîner une perte d'information . 1, 3, 4, 24, 45, 76

gamut ensemble des couleurs qu'un dispositif (écran) permet de reproduire. 46

GPU processeur graphique (*graphics processing unit*). 15, 16

KDE estimation par noyaux de la densité (*kernel density estimation*). Fournit une fonction estimant la densité de probabilité d'une variable aléatoire en tout point d'un domaine, à partir d'un échantillon. L'estimation dépend du choix du *noyau* et de la *fenêtre*, paramètre régissant le degré de lissage de l'estimation. Le noyau est souvent une fonction gaussienne standard . 19, 21, 48

marque élément visuel (ou graphique). 5, 7

partitionnement (*clustering*) partition de données calculée automatique en utilisant une notion de similarité entre éléments. 4, 15

SSIM Structural SIMilarity. 52, 61, 107

teinte (*hue*) couleur perçue, aussi appelé *ton*. Par exemple: "orange", "rouge", etc. 5, 6, 7, 9, 17, 26, 29, 39, 40, 42