Etude des projections de données comme support interactif de l'analyse visuelle de la structure de données de grande dimension

Nicolas Heulot

UNIVERSITÉ PARIS-SUD

ECOLE DOCTORALE INFORMATIQUE PARIS-SUD

LABORATOIRE ANALYSE DES DONNÉES
ET INTELLIGENCE DES SYSTÈMES
CEA LIST

DISCIPLINE : INFORMATIQUE

THÈSE DE DOCTORAT

Soutenue le 4 juillet 2014 par

Nicolas HEULOT

Etude des projections de données comme support interactif de l'analyse visuelle de la structure de données de grande dimension

Directeur de thèse :	M. Michaël AUPETIT	Ingénieur-Chercheur HDR (CEA LIST)
Co-directeur de thèse :	M. Jean-Daniel FEKETE	Directeur de Recherche (INRIA Saclay)
Composition du jury :		
Présidente du jury :	Mme. Michèle SEBAG	Directeur de Recherche (CNRS, Université Paris Sud)
Rapporteurs :	M. Gilles VENTURINI	Professeur (Université de Tours)
	M. Guy MELANCON	Professeur (Université Bordeaux 1)
Examinateur :	M. Renaud BLANCH	Maître de conférence (Université Grenoble 1)

Remerciements

Je tiens à remercier toutes les personnes qui m'ont accompagné et soutenu durant ces trois dernières années. En premier lieu, j'aimerai remercier mes directeurs de thèse, Michaël Aupetit et Jean-Daniel Fekete. Je les remercie tous les deux pour leur soutien indéfectible et ce dès les premiers jours. Michaël, merci pour ton enthousiasme, ta disponibilité au quotidien ainsi que l'autonomie que tu m'as accordé. Tu as toujours été présent pour me guider et me motiver à avancer jours après jours. Jean-Daniel, merci pour tes précieux conseils et ton aide dans les moments difficiles. Tu m'as montré la voie à suivre et nos échanges m'ont réellement aidé à prendre le recul nécessaire sur mes travaux de thèse.

Un grand merci ensuite à tous mes collègues de l'ex-LIMA pour toutes les pauses cafés et les pots de départ/arrivée que nous avons pu partager ensemble. Merci à tous les thésards du LADIS et d'AVIZ avec qui j'ai pu échanger des idées et partager des expériences. Un merci tout spécial à mes collègues de bureau et amis, Maxime, Florent et Jérémy. Je remercie également mes parents qui m'ont soutenus tout au long de mes études ainsi que mes amis proches qui m'ont permis de ne jamais abandonner.

Je remercie finalement les membres du jury, Michèle Sebag, Renaud Blanch, Guy Mélançon et Gilles Venturini qui m'ont fait l'honneur d'accepter d'évaluer mes travaux et dont les remarques ont contribué à améliorer ce mémoire.

Résumé

Acquérir et traiter des données est de moins en moins coûteux, à la fois en matériel et en temps, mais encore faut-il pouvoir les analyser et les interpréter malgré leur complexité. La dimensionnalité est un des aspects de cette complexité intrinsèque. Pour aider à interpréter et à appréhender ces données le recours à la visualisation est indispensable au cours du processus d'analyse. La projection représente les données sous forme d'un nuage de points 2D, indépendamment du nombre de dimensions. Cependant cette technique de visualisation souffre de distorsions dues à la réduction de dimension, ce qui pose des problèmes d'interprétation et de confiance. Peu d'études ont été consacrées à la considération de l'impact de ces artefacts, ainsi qu'à la façon dont des utilisateurs non-familiers de ces techniques peuvent analyser visuellement une projection.

L'approche soutenue dans cette thèse repose sur la prise en compte interactive des artefacts, afin de permettre à des analystes de données ou des non-experts de réaliser de manière fiable les tâches d'analyse visuelle des projections. La *visualisation interactive des proximités* colore la projection en fonction des proximités d'origine par rapport à une donnée de référence dans l'espace des données. Cette technique permet interactivement de révéler les artefacts de projection pour aider à appréhender les détails de la structure sous-jacente aux données. Dans cette thèse, nous revisitons la conception de cette technique et présentons ses apports au travers de deux expérimentations contrôlées qui étudient l'impact des artefacts sur l'analyse visuelle des projections. Nous présentons également une étude de l'espace de conception d'une technique basée sur la métaphore de lentille et visant à s'affranchir localement des problématiques d'artefacts de projection.

Mots clés : Visualisation d'information ; Fouille visuelle de données ; Données de grande dimension ; Projection de données

Abstract

The cost of data acquisition and processing has radically decreased in both material and time. But we also need to analyze and interpret the large amounts of complex data that are stored. Dimensionality is one aspect of their intrinsic complexity. Visualization is essential during the analysis process to help interpreting and understanding these data. Projection represents data as a 2D scatterplot, regardless the amount of dimensions. However, this visualization technique suffers from artifacts due to the dimensionality reduction. Its lack of reliability implies issues of interpretation and trust. Few studies have been devoted to the consideration of the impact of these artifacts, and especially to give feedbacks on how non-expert users can visually analyze projections.

The main approach of this thesis relies on taking these artifacts into account using interactive techniques, in order to allow data scientists or non-expert users to perform a trustworthy visual analysis of projections. The *interactive visualization of the proximities* applies a coloring of the original proximities relatives to a reference in the data-space. This interactive technique allows revealing projection artifacts in order to help grasping details of the underlying data-structure. In this thesis, we redesign this technique and we demonstrate its potential by presenting two controlled experiments studying the impact of artifacts on the visual analysis of projections. We also present a design-space based on the lens metaphor, in order to improve this technique and to locally visualize a projection free of artifacts issues.

Keywords : Information Visualization ; Visual Analytics ; High-Dimensional Data ; Multidimensional Scaling ;

Table des matières

Table des figures

1

Introduction

1.1 Motivation

Aujourd'hui acquérir et traiter des données est de moins en moins coûteux à la fois en matériel et en temps. Dans des domaines aussi variés que la génétique, les neurosciences, l'économie, l'imagerie, la sécurité et l'internet en général (streaming audio et vidéo, vente en ligne, réseaux sociaux), avec l'amélioration constante des capteurs et des performances des machines, ce sont de larges quantités de données qui sont traitées et analysées pour en extraire et exploiter très rapidement de l'information. Cependant seule une petite quantité d'information définie est pertinente au regard d'une tâche précise et à un instant donné. Pour obtenir cette quintessence à partir de masses de données issues de sources hétérogènes il faut repenser les méthodes actuelles de stockage, analyse et visualisation des données [82].

Les enjeux associés à cette évolution du rapport aux données se cristallisent dans différents domaines de recherche sous l'intitulé Big Data avec pour principales problématiques : le Volume, la Vélocité et la Variété. Les volumes de données se mesurent aujourd'hui en zettaoctets (10^{21}). Par exemple, les réseaux sociaux génèrent chaque jour des dizaines de téraoctets. Ces données arrivent de plus en plus par flux et nécessitent des traitements en temps réel pour permettre à de plus en plus d'entreprises de répondre de manière véloce aux besoins de processus métiers créateurs de valeur. Au delà du volume et de la vélocité, toutes ces données brutes ne sont pas toujours numériques, mais souvent de natures et de sources diverses (textes, images, signaux, etc.). Les valeurs de chaque instance de données, appelée *individu*, peuvent être catégorielles (comme des annotations de contenu sur une image), relationnelles (comme des connections entre machines dans un réseau), qualitatives (comme des notes sur un site de recommandation) ou quantitatives (comme des mesures issues de capteurs). Cette grande variété rend l'analyse plus complexe.

Un des aspects de cette complexité intrinsèque des données est l'augmentation de la dimensionnalité : les individus sont définis par des vecteurs numériques de très grande taille, où chaque composante correspond à une caractéristique, appelée *dimension* ou *variable* si l'on considère que les composantes sont dépendantes entre elles [263]. Les systèmes de recommandation sur les sites de vente en ligne comparent des centaines de caractéristiques produits. En neuroscience, l'électro-encéphalographie mesure l'activité électrique du cerveau sur des centaines d'électrodes. En analyse d'images, les comparaisons entre photos se font sur des centaines de milliers pixels. Dans ces données multidimensionnelles, dites de grande dimension, beaucoup de dimensions sont en réalité inutiles et les caractéristiques réellement pertinentes ne sont pas facilement identifiables.

Ces données de grande dimension doivent pourtant être analysées. L'analyse passe par la recherche de particularités, motifs, corrélations, dans le but de découvrir et caractériser un phénomène, formuler ou valider des hypothèses, élaborer des modèles à partir des données. Cette analyse se fait aussi bien à l'aide de méthodes automatiques que de techniques de visualisation, lesquelles permettent de mettre à profit les connaissances expertes d'êtres humains. L'interprétabilité des modèles et des relations entre données est également cruciale pour permettre à des non-experts en analyse de données de comprendre et exploiter l'information extraite à partir de données de grande dimension. Le recours à la visualisation est ainsi indispensable pour aider l'analyse et l'interprétation de ces résultats. Cette thèse adresse les enjeux de l'analyse de données de grande dimension et s'intéresse en particulier à la problématique générale :

- Comment visualiser des données de grande dimension ?

1.2 Contexte

L'analyse de données multidimensionnelles trouve ses origines dans les statistiques, où l'utilisation de visualisations sert à la fois à guider et transmettre le raisonnement analytique. Par exemple, l'analyse exploratoire de données introduite en statistiques par Tukey [232] repose sur la visualisation des données brutes sous la forme de nuages de points et d'histogrammes pour établir des hypothèses qui n'auraient pas été imaginées a priori. Ces hypothèses, qui seront ensuite validées (ou non) statistiquement par une analyse plus précise, sont suggérées par l'émergence d'informations particulières comme des tendances sur les valeurs des dimensions, des individus atypiques, appelés *outliers*, ou des motifs permettant d'associer des données similaires ensemble sous la forme de groupes, appelés *clusters*. En particulier, le partitionnement des données en clusters, appelé *clustering*, et l'analyse des caractéristiques de ces clusters (*Cluster Analysis*) est un problème central en fouille de données [219]. La fouille visuelle de données multidimensionnelles, par le biais de la visualisation des données brutes en matrice de nuages de points (SPLOM [55]) ou en coordonnées parallèles (PCP [119]), permet d'obtenir des intuitions sur le clustering (nombre de clusters, outliers, dimensions caractéristiques, validation d'étiquettes de classe).

Différentes méthodes automatiques et semi-automatiques existent également pour proposer des solutions au problème de clustering afin d'analyser les données par rapport à leur *structure sous-jacente*, c'est-à-dire en modélisant les caractéristiques de chaque cluster et les différentes relations entre ces clusters. Ces outils de traitement automatique viennent de domaines aussi variés que la reconnaissance de motifs, les statistiques, la recherche opérationnelle, l'apprentissage, ou la fouille de données [25]. Cependant il est parfois difficile d'inférer le cheminement fait par ces outils automatiques pour parvenir au résultat ; de plus leur efficacité est sensible aux caractéristiques intrinsèques des données. La visualisation permet à un analyste de données d'obtenir un aperçu des données, afin de se faire une idée sur l'approche automatique à utiliser et sur un paramétrage adapté. Le domaine récent de la Visualisation Analytique (*Visual Analytics*) se concentre sur l'étude de cette combinaison entre visualisation et méthodes automatiques d'analyse.

1.2.1 Visualisation Analytique

Le domaine de la visualisation d'information étudie les techniques permettant de construire des représentations visuelles de données abstraites selon un certain filtrage et niveau de détail, tout en mettant à profit les capacités humaines de perception. La visualisation d'information repose sur une approche de mise en forme des données selon leur structure (table, graphe, arbre, etc.) ayant pour vocation d'aider à interpréter les données [45]. Ce support est également interactif afin d'analyser visuellement et itérativement les données. La mantra de Shneiderman sur la recherche visuelle d'information donne un cadre à cette analyse visuelle : *"Overview first, zoom and filter, then details on demand"* [206]. Toutefois l'analyse visuelle atteint ses limites sur de grandes quantités de données complexes qui nécessitent le recours à des processus automatiques afin de synthétiser ou filtrer le nombre d'éléments à représenter. Par exemple, la sélection automatique de dimensions permet de concentrer l'analyse visuelle sur les caractéristiques les plus pertinentes. D'où l'interêt de combiner méthodes visuelles et méthodes automatiques pour exploiter au mieux les capacités des Hommes et des Machines.

La visualisation analytique se positionne au croisement de l'analyse automatique de données et de la visualisation d'information, afin de permettre de combiner l'expertise et les compétences humaines avec la précision et la rapidité des machines, comme défini par Keim [129] : *"Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets."* La visualisation sert ainsi de support interactif pour représenter et explorer les informations extraites par des outils d'analyse qui permettent de simplifier et résumer les données dans le but de les rendre exploitables. Ces outils d'analyse requièrent une gestion en temps réel des données afin de permettre une exploration réellement interactive par des analystes de données, des statisticiens, ou des experts dans un domaine précis. Dans le contexte de données de grande dimension, de nouvelles techniques de visualisation interactives sont ainsi développées pour aider à guider l'analyse afin de satisfaire une grande variété d'utilisateurs potentiels dans des domaines d'application comme la santé, la finance, le marketing ou la sécurité.

1.2.2 Visualisation par projection de données

La réduction de dimension par projection de données permet de visualiser en 2D (ou 3D) des données de grande dimension. Cette méthode représente les individus sous la forme d'un nuage de points où les individus les plus similaires sont proches entre eux sur la représentation et éloignés des individus qui leurs sont moins similaires. La projection de données nécessite donc l'introduction d'une relation de similarité ou de dissimilarité entre les individus, c'est-à-dire une mesure définissant les proximités entre individus dans l'espace des données (c'est-à-dire l'espace vectoriel des individus). L'objectif d'une mesure de similarité (ou de dissimilarité) est de mettre en évidence les ressemblances (et les différences) entre individus. La définition de cette mesure est un problème complexe qui dépend des données et du cas d'application ciblé. Par exemple, en analyse d'image, il existe de multiples mesures basées aussi bien sur l'intensité des pixels que sur les objets présents dans les images. Dans cette thèse, nous ne nous intéressons pas aux problématiques de construction d'une mesure de similarité et nous considérons des cas d'applications pour lesquels on ne cherche pas à remettre en cause cette mesure. On suppose que cette dernière est fournie a priori par un expert du domaine d'application et qu'elle met en évidence une structure sous-jacente aux données reflétant la réalité du point de vue de cet expert. Dans la suite, on appellera *proximités* les relations de similarité entre individus dans l'espace des données.

Une projection de données se lit comme une carte géographique et permet d'identifier des individus atypiques, représentés par des points qui tendent à être isolés, les principaux clusters, représentés par des amas de points disjoints les uns des autres, ainsi que la proximité entre ces clusters les uns par rapport aux autres, c'est-à-dire la topologie sous-jacente aux données (Figure 1.1). Il est à noter qu'une projection n'a pas d'axe (abscisse/ordonnée) directement interprétable. Seul l'agencement des points entre eux et plus précisément la distance 2D entre les points donnent son sens à la projection. Ceci constitue l'un des avantages de la visualisation par projection, c'est-à-dire son indépendance par rapport au nombre de dimensions dans les données. La projection permet ainsi de traiter des données de grande dimension en évitant les problèmes d'occlusions présents dans les visualisations de données multidimensionnelles comme les coordonnées parallèles, qui deviennent très difficiles à utiliser lorsque les données comportent plus d'une vingtaine de dimensions.

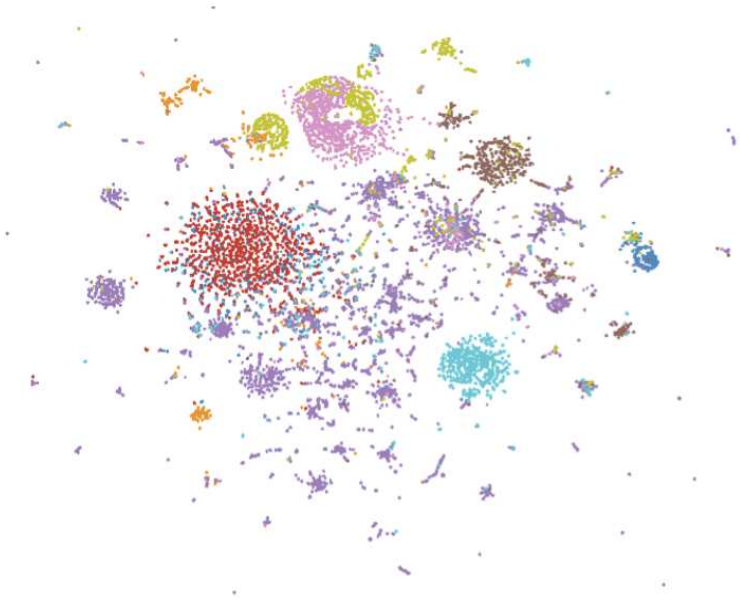


FIGURE 1.1 – Exemple de visualisation par projection d'une base de données de textes [116]. Chaque point correspond à un document et la similarité entre documents est construite à partir des mots qu'ils ont en commun. La projection met en évidence des amas de points caractéristiques de clusters de documents similaires ainsi que des points isolés caractéristiques d'outliers de données. La couleur des points correspond à un clustering automatique des données réalisé avant projection.

Néanmoins la projection de données pose d'autres problèmes spécifiques. En effet, elle ne préserve pas parfaitement les proximités d'origine entre individus. Trouver un positionnement 2D qui fait que les distances euclidiennes sur la projection correspondent parfaitement aux proximités d'origine n'est pas possible. Certains individus éloignés (respectivement proches) dans l'espace de grande dimension deviennent proches (respectivement éloignés) sur la représentation 2D. Ainsi même si les algorithmes de projection sont très efficaces pour préserver l'information de proximité entre individus, l'espace 2D n'est pas une représentation à l'identique de l'espace des données. Il nous apparait comme une représentation vue au travers d'un verre déformant. Ce problème de projection est bien connu des cartographes qui cherchent à projeter le globe terrestre sur un plan. Du fait de la réduction de dimension et de la topologie sphérique de la surface de la Terre, il est impossible d'obtenir une carte sans artefacts géométriques (Figure 1.2), c'est-à-dire une carte 2D qui respecte les distances mesurées sur le globe 3D. Les cartes géographiques sont sujettes à des compressions et étirements des distances d'origine sur le globe.

La déformation d'une projection de données se manifeste par la présence d'artefacts géométriques, mais également d'artefacts topologiques [14, 146]. Un cluster peut être découpé sur la projection 2D en plusieurs composantes non connexes, on parle alors d'artefact de déchirure. De plus, des clusters différents peuvent être projetés au même endroit en 2D, on parle d'artefact de faux voisinage. Ces artefacts topologiques interviennent à différents niveaux de granularité, selon que l'on considère des clusters ou que l'on se place au niveau des points entre eux. On peut également illustrer ces artefacts topologiques avec différentes projections 2D du globe terrestre (Figure 1.3).

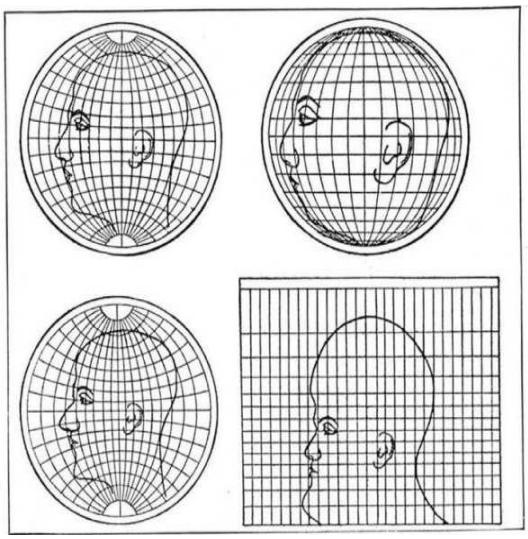


FIGURE 1.2 – Illustration provenant d'une édition du Scientific American de décembre 1921 et utilisant la tête d'un Homme pour montrer comment les projections cartographiques déforment les tailles et les formes. Le dessin en haut à gauche montre le globe 3D, ce qui explique pourquoi la tête n'est pas déformée. Les dessins en haut à droite, en bas à gauche et en bas à droite correspondent respectivement à une projection orthographique, stéréoscopique et une projection de Mercator.

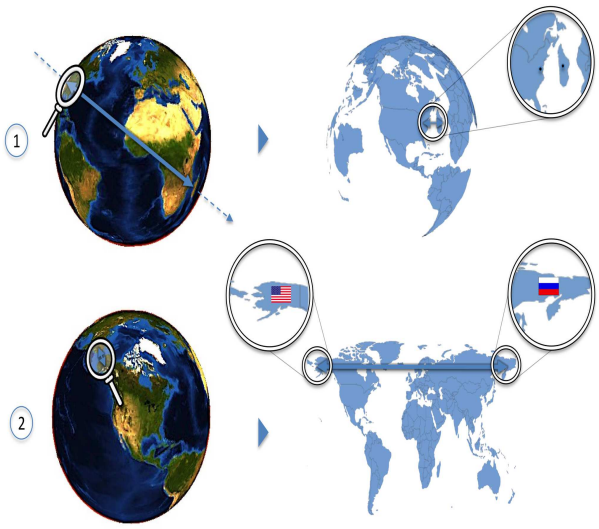


FIGURE 1.3 – Des artefacts de projection apparaissent lorsque l'on projette la Terre sur un plan 2D. (1) La projection orthogonale dévoile des faux voisinages : Washington se trouve juste à côté de Antananarivo. (2) La projection de Mercator met en valeur des déchirures : la Terre a été dépliée le long de l'équateur en une vue centrée sur l'Europe, représentant ainsi l'Alaska à l'opposé de la Russie et déchirant le détroit de Behring en deux. Ce même type d'artefacts topologiques existe lorsque l'on projette des données de grande dimension avec une technique de réduction de dimension.

Comme la métaphore “proche ≈ similaire” n’est pas toujours localement respectée, sans a priori sur les données, on ne peut pas faire entièrement confiance à la projection. La comparaison avec les cartes géographiques s’arrête là car lorsqu’on lit une carte géographique on a déjà une idée a priori du modèle 3D du globe terrestre. A l’image de la structure sous-jacente à des données qui se compose de clusters et d’outliers, la structure sous-jacente à ce modèle 3D correspond aux différents continents séparés par des océans dans lesquelles on trouve des îles. La connaissance de cette structure permet d’identifier les déchirures de continents sur une projection de Mercator ou toute autre projection du globe terrestre [241]. De plus, les projections avec des artefacts de faux voisinages ne sont pas utilisées en géographie, car les cartes doivent être localement fiables : deux points voisins sur la carte sont effectivement voisins sur le globe 3D.

1.2.3 Problématiques

Les artefacts sur les projections de données ne sont pas détectables s’il n’y a pas d’affichage d’informations supplémentaires relatives à ces distorsions. Par exemple, on remarque sur la Figure 1.1 que la couleur des points correspondant au clustering dans l’espace des données ne correspond pas exactement au clustering suggéré par le nuage de points 2D. Mais ceci ne permet pas pour autant de savoir si l’algorithme de projection est en cause ou bien si l’algorithme de clustering utilisé ne sépare pas aussi bien les clusters que la projection. Les artefacts topologiques nuisent à l’interprétabilité et à la confiance dans la représentation des données par projection [52]. Dans cette thèse, nous nous concentrons sur les aspects d’interprétation pour déterminer de quelle manière ces artefacts topologiques impactent l’analyse visuelle des projections

Le recours à l’affichage statique ou interactif d’informations sur ces artefacts peut permettre de compenser les biais introduits et ainsi permettre d’extraire de manière fiable la structure sous-jacente aux données à partir de la projection. Cependant peu d’études ont été consacrées à la considération de ces artefacts de projection et à la façon dont des utilisateurs, non nécessairement familiers des projections, peuvent appréhender ce problème afin de correctement analyser visuellement une projection de données.

Cette thèse s’intéresse en particulier aux questions suivantes :

- Quel est l’impact des artefacts de réduction de dimension sur l’analyse visuelle des projections de données ?
- L’interactivité peut-elle permettre de s’affranchir de ces problèmes afin d’exploiter au mieux le potentiel des projections ?
- Indépendamment du niveau d’expertise de l’utilisateur, peut-on utiliser les projections pour appréhender la structure sous-jacente à des données de grande dimension ?

1.3 Approche

L'approche soutenue dans cette thèse repose sur la prise en compte interactive des artefacts afin d'aider l'analyse visuelle des projections de données, c'est-à-dire améliorer la fiabilité des interprétations faites sur les données à partir de la projection. En particulier, nous reprenons le concept de la *visualisation interactive des proximités* [14] afin de mettre en évidence les artefacts. Cette technique permet de visualiser, directement sur la projection par le biais de la couleur, les proximités d'origine entre individus par rapport à une référence sélectionnée par l'utilisateur (Figure 1.4). Nous étudions également les possibilités offertes par cette approche pour des utilisateurs non nécessairement familiers des projections de données au travers d'une taxonomie des tâches d'analyse visuelle des projections.

Les projections de données sont aujourd'hui principalement utilisées par des analystes de données selon deux contextes d'application :

Dans un *contexte exploratoire*, c'est-à-dire sans modèle a priori de la structure sous-jacente aux données, les analystes cherchent à identifier visuellement des clusters et détecter des outliers dans les données. Les outliers identifiés peuvent correspondre à des anomalies, dont il est intéressant de trouver l'origine pour y remédier. Enumérer les clusters permet de paramétrer ensuite un algorithme de clustering automatique pour classifier les données.

Dans un *contexte confirmatoire*, les analystes considèrent une connaissance a priori de la structure sous-jacente aux données, c'est-à-dire une partition des données en classes qui associe une étiquette à chaque individu. Ces classes modélisent des clusters dans les données et peuvent être issues d'un modèle du phénomène observé connu par ailleurs. Au regard de ces classes, l'objectif des analystes est alors de valider si de nouvelles données satisfont le modèle ou non. Les analystes étudient la projection relativement aux étiquettes pour déterminer la proximité entre les classes, leur chevauchement éventuel et détecter des outliers de classes, c'est-à-dire des individus appartenant à un cluster composé majoritairement d'individus d'une autre classe.

Si aujourd'hui les projections sont principalement utilisées par des experts en analyse de données, la métaphore de proximité, sur laquelle les projections reposent, est relativement intuitive [134]. On peut trouver différents cas d'applications pour lesquelles les projections de données peuvent servir à des non-experts, c'est-à-dire des utilisateurs ne souhaitant pas extraire des informations relatives à la classification des données mais plutôt exploiter directement ces informations pour effectuer par exemple de la recherche d'information ou contrôler un système.

Dans un cas d'application de contrôle de système, la tâche confirmatoire, de validation des données par rapport à un modèle, peut être effectuée par des non-experts, afin de détecter visuellement un écart du système par rapport à son modèle initial. La projection permet alors de visualiser directement les informations relatives aux anomalies et de sélectionner les éléments atypiques (outliers, clusters non fidèles aux classes). Par exemple, ce scénario peut s'appliquer au suivi des profils de visiteurs d'un site web afin de détecter l'émergence d'un nouveau type de profil de visiteur, pour ensuite permettre d'aider à le caractériser. L'étude locale des proximités entre classes sur la projection peut également être effectuée par des non-experts. Appliquée à l'analyse de portefeuilles boursiers, cette tâche permet de détecter si les actifs d'un portefeuille d'actions d'un client potentiel sont trop risqués, car trop similaires à des actions de référence identifiées comme potentiellement dangereuses.

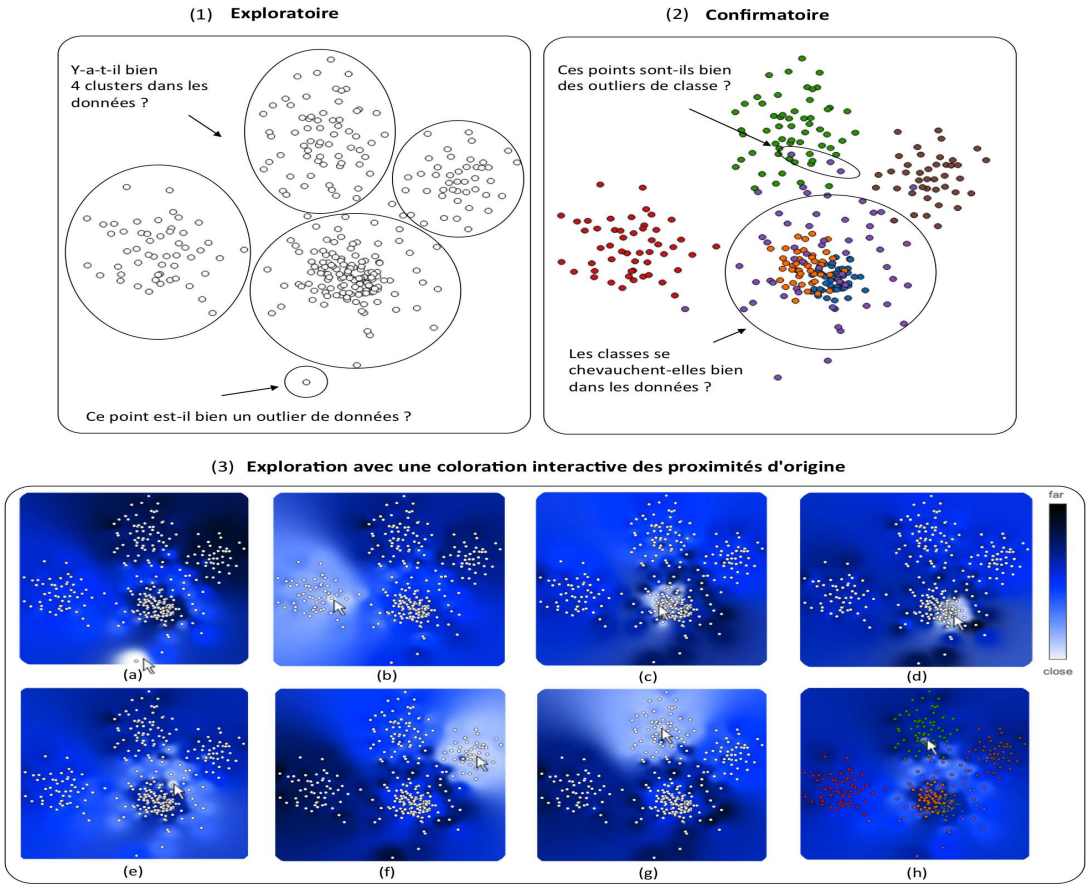


FIGURE 1.4 – Cette figure présente les principales questions, liées à la fiabilité de la projection, qu'un analyste de données doit se poser lors de son analyse visuelle, dans un contexte exploratoire ou confirmatoire (1,2). L'exploration de la projection avec la coloration interactive des proximités d'origine permet d'obtenir une réponse fiable à chacune de ces questions (3). Dans la suite (chapitre 2, section 2.4), nous proposons une taxonomie des tâches d'analyse visuelle qui formalise ces questions. Le point de référence de la coloration interactive est celui le plus proche du curseur de la souris.Plus la couleur est claire et plus la distance à la référence dans l'espace des données est faible, autrement dit plus les individus sont similaires à la référence. Les sauts dans les différentes colorations de la projection en nuances de bleu montrent qu'il y a 6 clusters sous-jacents aux données (b,c,e,f,g,h). On observe que le point en bas de la projection (1) est effectivement un outlier de données (a), car il est distant du reste des données (les couleurs des autres points étant très sombres). On remarque également que les 3 points en haut de la projection (2) sont effectivement des outliers de classe (h). Cette projection a été obtenue à partir d'une Analyse en Composantes Principales [170] d'un jeu de données synthétique composé de 6 clusters générés à partir de Gaussiennes aléatoirement distribuées dans un espace à 10 dimensions. Cette exemple servira de fil-rouge dans la suite.

Dans un cas d'application plus exploratoire, la projection peut également servir à des non-experts comme support interactif pour naviguer de manière structurée dans les données. Afin d'extraire des caractéristiques visuelles ou de détecter d'éventuelles erreurs d'acquisition dans une collection d'images issues par exemple d'instruments de microscopie, l'exploration de la collection est un travail fastidieux qui doit souvent être effectué rapidement. La projection peut alors être utilisée comme support pour organiser le travail d'exploration et guider l'utilisateur, à la manière d'une carte, vers des images atypiques.

Cependant sans informations relatives aux artefacts de la projection, les non-experts, comme les analystes de données, ne peuvent pas savoir si la projection représente fidèlement l'espace des données et si les inférences qu'ils font sur les données à partir de la projection sont viables dans la réalité. Notre approche part du constat qu'on ne peut pas exploiter une projection de manière fiable sans prendre en compte ses artefacts. Nous faisons l'hypothèse forte que l'on peut considérer une mesure de similarité révélant une structure sous-jacente aux données qui reflète la réalité. Nous prenons le parti d'utiliser les projections augmentées d'outils interactifs permettant de faire fi des artefacts, afin que n'importe quel utilisateur, souhaitant extraire ou utiliser la structure sous-jacente aux données, puisse utiliser une projection indépendamment de l'algorithme l'ayant généré et de sa qualité en termes de distorsions.

1.4 Contributions

Les différentes contributions de cette thèse sont axées sur l'implémentation et la validation de cette approche d'un point de vue utilisateur. Différentes méthodologies ont été mises en oeuvre pour parvenir à ces fins : la conception et l'implémentation de techniques, la réalisation d'expérimentations contrôlées et une étude d'espace de conception (design space). Les principales contributions de cette thèse se décrivent comme suit et résument la démarche de recherche entreprise :

❶ Nous avons introduit un cadre théorique avec une taxonomie des tâches de l'analyse visuelle des projections de données mettant en évidence les risques d'erreurs d'interprétation associés aux différents types d'artefacts et nous avons également positionné le biais introduit par les artefacts de projection dans la chaîne de traitement par réduction de dimension afin, d'une part, d'en souligner l'importance et, d'autre part, de mettre en évidence un manque d'outils, dans la littérature, permettant de s'en affranchir.

❷ Nous avons mis en place un cadre expérimental, avec deux expérimentations contrôlées, afin de quantifier l'impact des artefacts sur l'analyse visuelle des projections et d'évaluer les performances, pour l'aide à l'analyse visuelle des projections, de techniques comme la *visualisation interactive des proximités* [14] pour laquelle nous avons fait évoluer la conception.

❸ Nous avons étudié l'espace de conception d'une technique interactive, basée sur la métaphore de lentille, afin de définir des éléments de concept et des critères permettant de résoudre des problèmes d'encodage graphique et d'interaction, liés aux artefacts de faux voisinages, dans le cadre de l'aide à l'analyse visuelle des projections.

1.5 Plan du manuscrit

Le chapitre 2 décrit les travaux reliés à la fouille visuelle de données par réduction de dimension. Nous donnons, dans un premier temps, un aperçu des techniques de visualisation de données multidimensionnelles, avant de décrire les différentes méthodes de projection, ainsi que les mesures de qualité et les techniques de visualisation associées. Par la suite, nous explicitons le pipeline de réduction de dimension ainsi que les différents biais introduits à chaque étape. Après une présentation des systèmes d'analyse exploratoire implémentant ce pipeline, mais ne fournissant pas d'outils pour prendre en compte les artefacts de projections, nous présentons les cas d'utilisation des projections. Nous introduisons finalement une taxonomie des tâches d'analyse visuelle des projections, explicitant les possibles biais dus aux artefacts de projections.

Le chapitre 3 présente les enjeux associés à la *visualisation interactive des proximités* [14]. Nous revisitons la conception de cette technique, que nous appelons désormais ProxiViz, au niveau de l'encodage graphique et de l'interaction de navigation.

Le chapitre 4 présente une première expérimentation contrôlée qui a été réalisée, pour une tâche de clustering visuel, de manière à confronter ProxiViz à l'état de l'art, à savoir la projection sans ajout d'information et la projection avec une coloration des zones de distorsion. Cette expérience permet également de comparer différents encodages visuels de la technique. Nous rapportons et discutons ensuite les résultats.

Le chapitre 5 présente une seconde expérimentation contrôlée qui a été réalisée pour étudier les performances de la projection sans ajout d'information et de ProxiViz, par rapport à différentes tâches d'analyse visuelle (Figure 1.4). Cette expérience permet également de quantifier l'impact des artefacts sur la précision des analyses visuelles. Nous rapportons et discutons ensuite les résultats.

Le chapitre 6 introduit une étude de l'espace de conception d'une lentille basée sur ProxiViz et permettant de nettoyer localement la projection de ses artefacts de faux voisinages. Cette étude vise à résoudre les problématiques de ProxiViz, liées aux artefacts de faux voisinages, sur la représentation des proximités d'origine, la navigation avec la technique et l'extraction des structures sous-jacentes aux données par brossage 2D sur la projection. Nous introduisons ensuite une implémentation de ce concept de lentille, nommée ProxiLens, dont nous illustrons la portée, avec un jeu de données d'images, sur les différentes tâches d'analyse visuelle des projections.

Nous concluons finalement, dans le chapitre 7, sur les différentes contributions ainsi que les perspectives de poursuite de ce travail. Ces travaux ont donné lieu à des publications que nous listons avant la section bibliographique.

2

Etat de l'art

Dans ce chapitre, nous discutons les principaux travaux liés à notre sujet de recherche : les problématiques associées à la visualisation de données multidimensionnelles, l'intérêt des projections pour visualiser des données de grande dimension, puis comment évaluer la qualité de celles-ci et analyser visuellement la structure sous-jacente aux données.

Après avoir abordé la littérature de la visualisation de données multidimensionnelles et expliqué ses limitations en termes de dimensionnalité, nous donnons un aperçu des techniques de projection par réduction de dimension. Nous présentons ensuite les mesures permettant d'analyser leur qualité ainsi que les techniques permettant de visualiser ces mesures. Puis nous introduisons le pipeline de réduction de dimension en prenant en compte les biais introduits à chaque étape et enfin nous expliquons les défis de l'analyse visuelle des projections.

2.1 Visualisation d'information

"Information visualization is the use of computer-supported, interactive, visual representations of abstract data to amplify cognition" Card et al. [45]. La visualisation d'information vise à créer des représentations visuelles de données abstraites associées à des techniques d'interaction, dans le but de donner un aperçu des données, aider à les comprendre, valider des intuitions ou découvrir des choses inattendues. Une visualisation représente les données de manière à tirer profit des capacités du système visuel humain. Afin d'être interprétable et "efficace", elle doit également s'adapter à son contexte d'application, c'est-à-dire à ses utilisateurs et aux tâches d'analyse qu'ils souhaitent réaliser. L'efficacité mesure ce qu'apporte la visualisation à la réalisation d'une tâche d'analyse. Cependant elle est difficile à quantifier [240], de même que le coût d'une visualisation. Ce coût s'exprime souvent en termes de temps de réponse, de précision dans la réponse, ou par rapport à l'espace visuel exploité. L'utilisation de la visualisation d'information permet par exemple de révéler des anomalies ou outliers (c'est-à-dire des données qui n'ont pas les mêmes caractéristiques que le reste des données), trouver des clusters (c'est-à-dire des données similaires qui ont suffisamment de caractéristiques communes pour indiquer la présence d'un phénomène structurant) ou mettre en valeur des tendances (c'est-à-dire des données qui évoluent de la même manière et décrivent ainsi un phénomène prévisible).

Le processus de visualisation d'information s'inscrit généralement dans un cadre plus large d'analyse exploratoire, ou d'extraction de connaissances. On distingue principalement trois contextes d'utilisation de la visualisation d'information [248] :

- explorer les données : on cherche à découvrir et extraire des informations "intéressantes", sans a priori sur les données (hypothèses, modèle), pour susciter des interrogations et suggérer des hypothèses.
- analyser les données : on cherche à répondre à des questions, vérifier des hypothèses, valider un modèle a priori.
- présenter les données : on cherche à communiquer des observations ou modélisations qui résultent d'une analyse préalable pour laquelle on souhaite mettre en valeur les conclusions.

Le processus qui génère une image à l'écran à partir des données permet ensuite de construire, valider ou présenter un modèle des données. Il peut être décrit comme un pipeline composé de différentes étapes, chacune pouvant s'adapter par le biais de l'interaction aux besoins de l'utilisateur. Les données brutes sont transformées, filtrées, puis associées à des représentations visuelles qui seront coordonnées et mises en forme pour être visualisées et analysées à l'écran [45].

Construire une visualisation de données peut aussi être vu comme chercher une codification visuelle de l'information. Cet encodage graphique associe des ensembles de valeurs à différentes variables visuelles. Ces variables ont une qualité perceptuelle (position, taille, couleur, forme, texture, orientation, intensité [120] ou surface, mouvement, texte [256]) plus ou moins appropriée selon la nature des valeurs encodées. Par exemple, la couleur est perceptuellement plus efficace que les autres variables pour représenter des valeurs qualitatives. Les travaux issus des Sciences Cognitives apportent un socle pour essayer de tirer profit au mieux des propriétés de la perception humaine et ainsi mettre au mieux en valeur l'information. La prise en compte du processus de perception pré-attentive ou les lois de la Gestalt [134] fournissent par exemple des recommandations pour construire des représentations efficaces. Au delà des différentes étapes qui permettent

d'aboutir à une représentation graphique, la conception d'une visualisation commence par la prise en compte de la nature des données brutes.

On peut distinguer les données selon leur nature, c'est-à-dire selon qu'elles se présentent sous forme de table de données, dite structurée, ou sous forme de collection d'objets (corpus de textes, base d'images, ensemble de signaux). Les données non-structurées sont difficiles à représenter car chaque objet forme un tout et il faut introduire des mesures pour pouvoir les comparer à partir de leur contenu. Considérant des données structurées sous forme tabulaire, on compte principalement 7 types de données [206] : 1D, 2D, 3D, temporelles, multidimensionnelles, arbres, graphes. Les données tabulaires multidimensionnelles sont considérées de grande dimension à partir d'une vingtaine de dimensions. Ceci correspond à la limite d'efficacité des visualisations de données multidimensionnelles n'utilisant pas la réduction de dimension. Toutefois aucune définition précise n'existe dans la littérature.

Différentes techniques permettent de visualiser une collection d'objets. Par exemple, les graphes sous forme noeuds et liens permettent de représenter les relations d'association qu'entretiennent les objets les uns par rapport aux autres. La structure d'une collection d'objets se définit au travers d'une mesure de similarité préalablement choisie sur les données. Différentes mesures de similarité existent selon le type d'objet considéré et la sémantique de comparaison que l'on souhaite établir pour répondre à une tâche précise, c'est-à-dire le sens sous-jacent à la mesure de similarité et permettant son interprétation. On peut ainsi distinguer les relations de similarité implicites obtenues par une métrique de distance, des relations explicites qui constituent les données elles-mêmes [68]. Sur les réseaux sociaux par exemple, on peut comparer des profils d'utilisateurs selon leurs caractéristiques par le biais d'une mesure de similarité, c'est-à-dire d'une relation implicite de similarité, ou bien on peut comparer les profils selon leurs relations, c'est-à-dire des liens explicites d'association. Nous nous intéressons dans cette thèse à des données de grande dimension, c'est-à-dire une catégorie de données multidimensionnelles sous forme tabulaire où les individus ont plus d'une vingtaine de dimensions et à des objets complexes ou collections d'objets pour lesquels une mesure de similarité existe ou est prédéfinie.

2.1.1 Visualisation de données multi-dimensionnelles

Les données multidimensionnelles se définissent comme un ensemble d'individus ou observations X, où l'individu x_i est un vecteur de m attributs $x = (x_{i,1}, ..., x_{i,m})$. Considérant des attributs numériques $x_{i,j} \in \mathbb{R}$, on parle de données multidimensionnelles si les attributs sont indépendants (dimensions) et de données multi-variées si les attributs sont dépendants (variables) [263]. Dans la suite, nous utilisons par défaut le terme *dimension* et le terme *variable* si on suppose une dépendance entre les attributs, comme la corrélation. Mais par convention, nous utiliserons le terme *données multidimensionnelles* plutôt que *données multidimensionnelles multi-variées*, même si cette terminologie est plus précise car elle ne sous-entend pas que l'on connait a priori les relations entre attributs.

L'ensemble des attributs peut être homogène si tous les attributs sont de même nature, ou hétérogène dans le cas contraire. On distingue les attributs avec des valeurs numériques (données ordinales) et ceux avec des valeurs non-numériques (données nominales). Plus précisément, on peut distinguer parmi les données ordinales celles qui sont binaires, discrètes (qualitatives), ou

continues (quantitatives). On peut ensuite distinguer parmi les attributs différentes métadonnées qui décrivent le contenu des données (étiquette de classe, unité de mesure, identifiant). Des attributs synthétiques peuvent également être dérivés par calcul pour représenter des valeurs sur un échantillon particulier (comme une moyenne ou un écart type) ou bien pour combiner différents attributs de départ (comme les composantes principales). Des attributs peuvent être extraits automatiquement pour caractériser le contenu des données, comme des caractéristiques d'images [95] ou de signaux [25].

Les dimensions spatiales et temporelles sont très importantes, car directement interprétables. Une trajectoire est un exemple de donnée spatio-temporelle. Ces deux dimensions définissent un cadre spécifique de tâches d'analyse et de navigation [10]. Prises séparément, leur encodage visuel spécialise la représentation (pour plus de détails voir l'état de l'art sur la visualisation de données temporelles [4]). La prise en compte de ces dimensions est également primordiale en visualisation de données scientifiques, qui est un domaine connexe à la visualisation d'information. Dans ce domaine, les données multi-variées se composent de champs de scalaires, de vecteurs ou de tenseurs (voir l'état de l'art de la visualisation scientifique [90]). Dans cette thèse, nous nous intéressons à des données multidimensionnelles composées de valeurs numériques et sans composantes temporelles ou spatiales.

La taxonomie des tâches bas niveau d'analyse visuelle [8] donne un aperçu des différents critères permettant de comparer les forces et faiblesses d'une technique de visualisation et des interactions avec un système : trouver une valeur dans un intervalle, filtrer les individus, calculer des valeurs dérivées sur un échantillon (moyenne, écart type), trouver un extremum sur un intervalle de valeurs, trier les individus, déterminer une plage de valeurs, caractériser une distribution, trouver des anomalies, définir une partition des données, trouver des corrélations entre variables. Dans la suite, nous noterons clustering la tâche de partionnement des données en clusters, c'est-à-dire en groupes de données similaires entre elles.

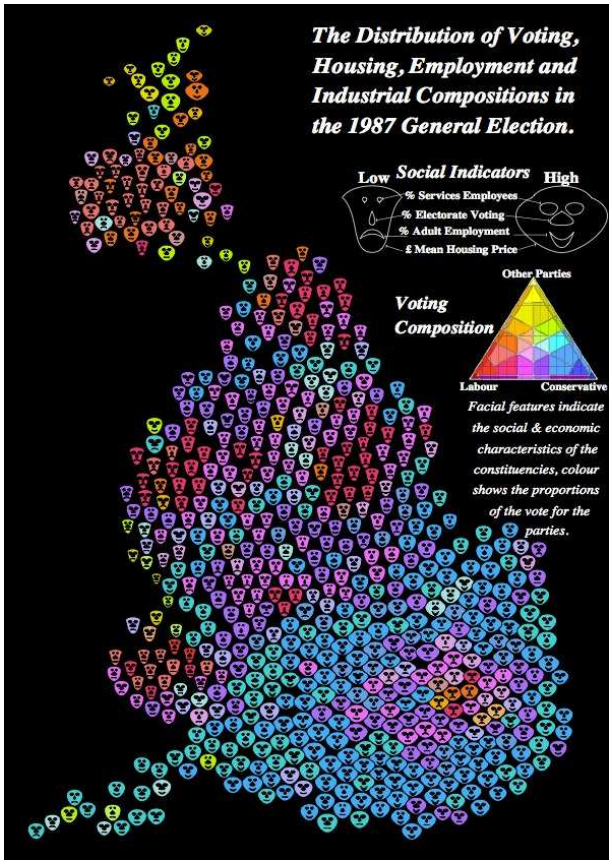
Il existe de nombreuses représentations graphiques usuelles de données multidimensionnelles [120, 54, 212] sous forme par exemple d'un nuage de points, d'une courbe, de barres, de piles, etc.. On peut catégoriser ces représentations selon le type de primitives graphiques utilisées (point, ligne, région, mélange de primitives) ainsi que le positionnement des primitives dans l'espace 2D ou 3D (repère cartésien ou polaire).

Mais ces représentations sont limitées par le nombre de dimensions prises en compte. Généralement en 2D, on compte une dimension pour chaque axe (x/y) du repère cartésien et une dimension par variable visuelle supplémentaire. Par exemple, la couleur ou la taille dans un nuage de points encodent respectivement des dimensions qualitatives ou quantitatives. Plusieurs variations des représentations standards 2D/3D, comme les graphiques à multiples courbes ou les matrices de permutation de Bertin [120], ont amorcé le design de nouvelles techniques de visualisation.

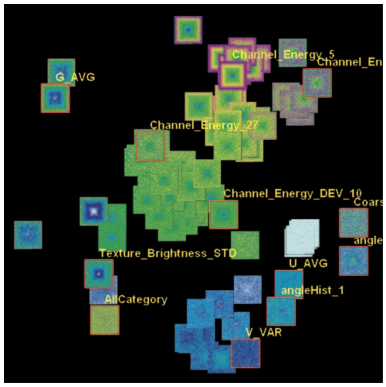
Différents états de l'art [132, 263, 45, 107] catégorisent les principales approches existantes de visualisation d'information. Keim [128] classifie ces techniques selon le type de données à visualiser (1D, 2D, multidimensionnel, graphes / hiérarchies, textes et algorithmes), la technique de représentation (diagramme 2D/3D, projection géométrique, basé sur les icônes, orienté pixel, hiérarchique) et le type d'interactions (projection, filtrage, zoom, distorsion, "link & brush"). Cette classification a ensuite été reprise et appliquée au contexte de l'analyse visuelle de données [84]. Dans le cadre de données multidimensionnelles, nous nous intéressons principalement à trois stratégies de représentation : basée sur les icônes, orientée pixel, par projection géométrique. Nous reprenons ci-dessous quelques techniques pour illustrer chaque stratégie, sans être exhaustif.

Représentation basée sur les icônes/glyphes Les représentations iconographiques représentent les individus sous forme de glyphes dont les caractéristiques de forme dépendent des valeurs prises sur les dimensions. Les visages de Chernoff [97] associent deux dimensions à la position du visage dans un repère cartésien et le reste des dimensions paramétrisent les propriétés du visage : forme, nez, bouche, yeux. Cependant ces variables visuelles ne sont pas directement comparables. DICON [44] permet de visualiser le clustering sous forme d'icone, avec une représentation "treemap" affichant les valeurs de chaque dimension. La forme de l'icone donne également des informations statistiques sur chaque cluster (taille du cluster, kurtosis). Cette représentation permet de comparer des clusters et interpréter leurs caractéristiques. Des nombreuses extensions de ces représentations par glyphes ont été développées et ont récemment été formalisées [29]. Ce type de représentation permet d'identifier assez facilement des outliers ou de comparer des clusters deux à deux, mais il est plus difficile d'obtenir des informations comme la corrélation ou le clustering total des individus.

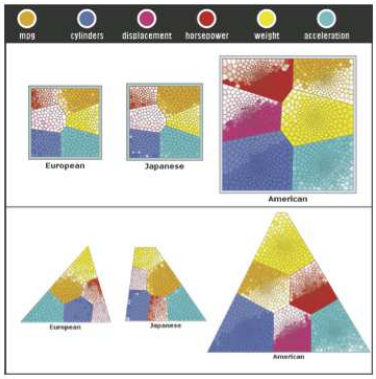
Représentation orientée pixels Les représentations orientées pixels représentent les valeurs des individus pour une dimension donnée par le biais d'une échelle de couleur [132, 131] (Figure 2.1). Les intervalles de valeur dépendent de l'échelle de couleur choisie, mais cette technique permet clairement de représenter un très grand nombre d'individus. Différentes configurations sont possibles pour ordonner les individus (en spiral, selon un axe, récursivement) par rapport à leur distance à la requête. La couleur permet ensuite de distinguer les clusters selon une dimension ainsi que les corrélations entre deux variables. Cette représentation peut également être utilisée dans un contexte de requête, avec le système VisDB [130] par exemple. La distance par rapport à l'individu requête est directement affichée. On peut également l'utiliser avec un positionnement des dimensions par projection en 2D à partir d'une mesure de corrélation, comme dans VaR [268, 267] par exemple. Cette technique permet de visualiser les relations entre des centaines de dimensions et de visualiser les valeurs pour un très grand nombre d'individus afin de détecter des outliers ou des tendances. Mais comme pour toutes les techniques orientées pixels, le clustering total des individus est plus difficile à identifier car il faut mentalement assembler le clustering présent sur chaque dimension.



(a) Visages de Chernoff



(b) VaR



(c) DICON

FIGURE 2.1 – Représentations par glyphes et orientées pixels. (a) Illustration des visages de Chernoff sur un cartogramme de Dorling [69] des élections générales de 1987 au Royaume-Uni. Sur cette représentation géographique de la Grande Bretagne, chaque région est représentée par un visage de Chernoff indiquant le nombre de votants (nez), le prix moyen du logement (visage), le pourcentage d'emplois (bouche), le pourcentage d'employés dans les services (yeux). La couleur indique la proportion de vote pour chaque parti politique. (b) Représentation Valeur/Relation (VaR) avec les différentes dimensions positionnées par projection MDS d'un jeu de données de 89 attributs visuels sur 10,417 segments d'image [267]. (c) Représentation DICON [44] d'un jeu de données de 407 voitures décrites selon 7 attributs. On observe que le cluster Etats Unis est 3 à 4 fois plus gros que les deux autres et que les clusters Europe et Japon sont similaires même si leurs distributions sont différentes. Les voitures Américaines comparées aux Européennes et Japonaises sont plus lourdes (en jaune), avec une plus grosse cylindrée (en violet), avec plus de cylindres (en bleu), une accélération plus faible (en cyan), plus de chevaux (en rouge) et moins de miles-par-gallon (en orange).

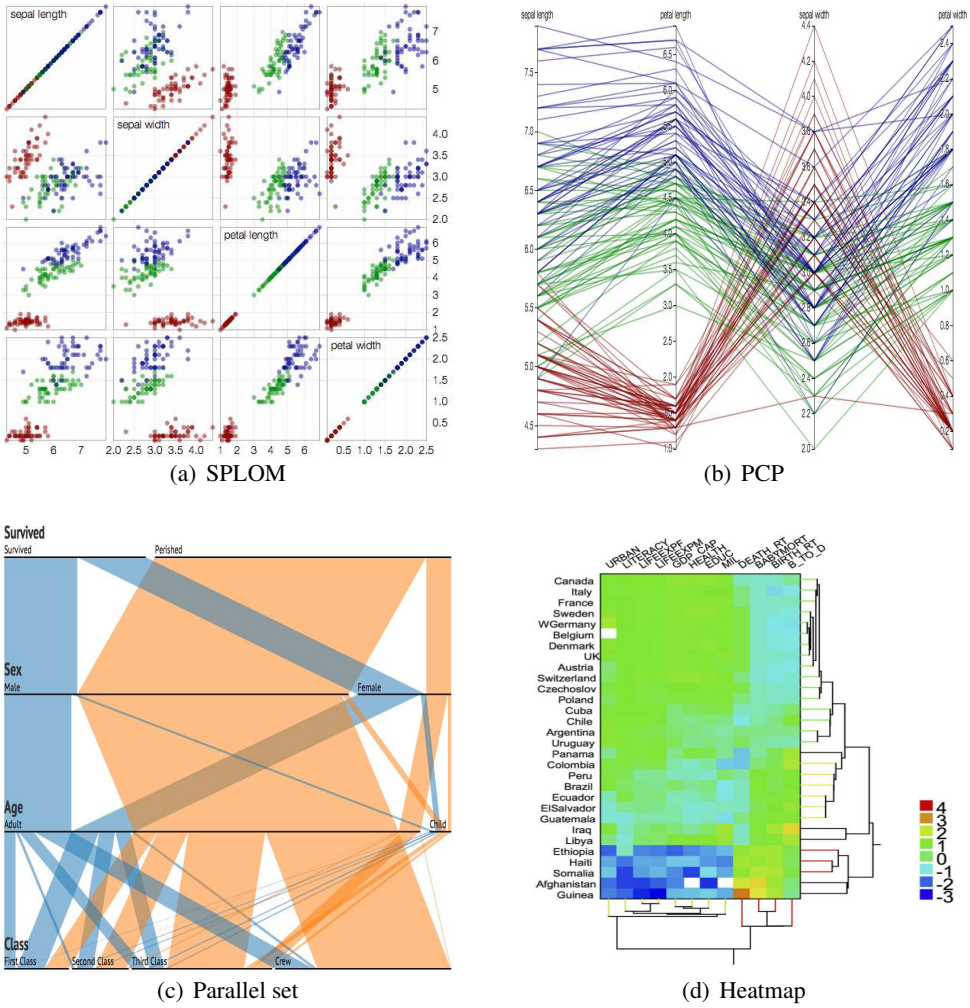
Représentation par transformation géométrique des axes Chaque individu est projeté selon une transformation géométrique 2D des axes correspondant à chaque dimension (Figure 2.2). On peut différencier les représentations par transformation géométrique selon leur repère de coordonnées : cartésien ou polaire. On peut également distinguer les représentations selon les primitives géométriques utilisées :

Points : Au delà des représentations usuelles en nuage de points 2D ou 3D, on trouve les matrices de nuages de points (Scatterplot Matrix - SPLOM) qui affichent sous forme d'un tableau lignes/colonnes un nuage de points pour chaque combinaison de paires de dimensions [54]. L'ordre des lignes et colonnes est le même et chaque paire de dimensions est représentée deux fois à cause de la symétrie diagonale. La diagonale peut servir à afficher les noms des dimensions ou bien à représenter leur distribution par le biais d'un histogramme. Cette technique permet de clairement détecter les tendances parmi les individus et les corrélations entre variables. La plupart des autres représentations par points utilisent des transformations par projection des axes.

Lignes : Les coordonnées parallèles affichent chaque dimension normalisée linéairement selon des axes parallèles espacés régulièrement les uns par rapport aux autres et chaque individu est représenté par une ligne qui coupe chaque axe au niveau de la valeur de l'individu pour la dimension associée à l'axe [119]. L'ordre des axes est important car la corrélation entre variables ne peut être facilement révélée qu'entre deux axes consécutifs. Les coordonnées parallèles ont fait l'objet de beaucoup d'extensions et améliorations [103] mais cette technique ne fonctionne que pour des données ordinales ou quantitatives. Elle a été étendue aux données nominales avec les ensembles parallèles (parallel sets) [20]. Un positionnement radial des axes est également possible. Sur le même modèle que les coordonnées parallèles, les représentations par points et par lignes peuvent également être mêlées et le positionnement des axes est généralisable selon différents motifs [53].

Regions : Les représentations en matrices utilisent des régions pour représenter les données en faisant varier la taille des cellules comme les survey plots [98] ou en faisant varier leur couleur. Dans ce dernier cas, on parle de *heatmap* [258] (en particulier pour la représentation de puces ADN), de matrice de permutation [120], ou juste de matrice [265]. Le système TableLens [107] combine ces techniques selon différents niveaux de détails. Différentes interactions permettent d'explorer le tableau de données selon de multiples ordonnancements de lignes et colonnes afin de mieux détecter des tendances, des corrélations ou des outliers. La représentation par empilement des dimensions [141] (dimensional stacking) utilise également une représentation en régions rectangulaires où chaque paire de dimensions est récursivement imbriquée dans le système de coordonnées d'une autre paire de dimensions englobantes. Les dimensions doivent être discrétisées et la hiérarchie d'imbrication des dimensions doit être choisie de telle sorte que les dimensions les plus pertinentes englobent les moins intéressantes. Cette représentation permet de visualiser le vide dans les espaces de grande dimension.

Toutes ces techniques permettent de révéler la distribution des données ainsi que les intervalles de valeurs pris par les individus sur chaque dimension. On peut également observer des relations de corrélations entre variables ou des tendances parmi les individus. Mais la clarté de ces représentations est directement liée au nombre de dimensions. Des problèmes de chevauchement et d'occlusion peuvent apparaître lorsque le nombre de dimensions est trop important, ce qui nuit à l'interprétation et l'interaction avec ces visualisations par transformation géométrique.



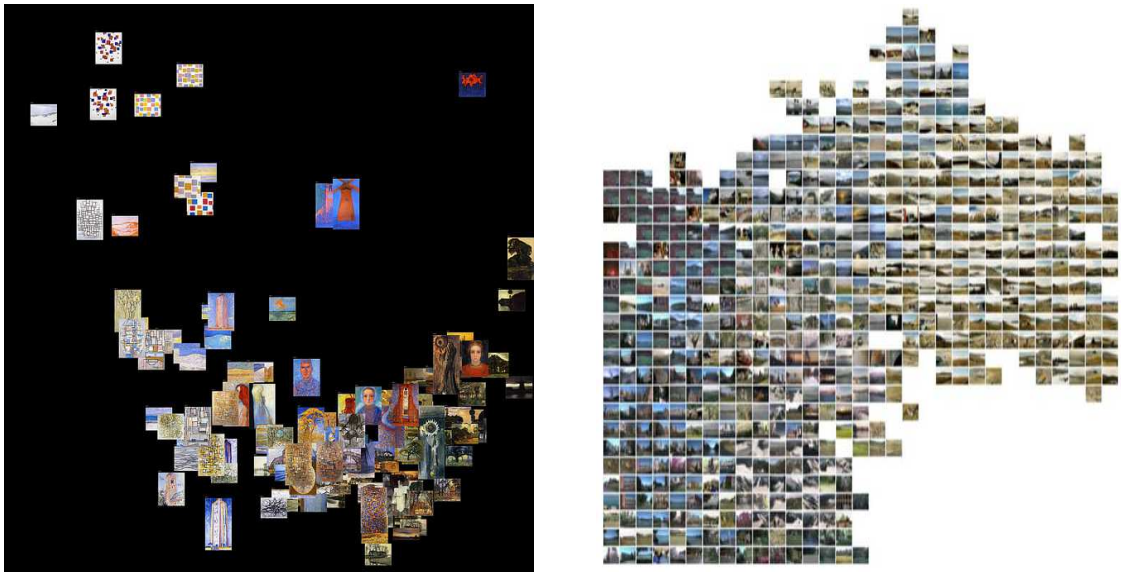
(a) SPLOM (b) PCP

(c) Parallel set (d) Heatmap

FIGURE 2.2 – Représentations par transformation géométrique des axes. (a) Représentation en matrice de nuage de points (SPLOM) du jeu de données IRIS composé des attributs de taille et d'épaisseur des pétales et sépales sur 150 individus répartis en 3 types d'iris : Setosa (rouge), Versicolour (vert) et Virginica (bleu). (b) Représentation en coordonnées parallèles (PCP) du jeu de données IRIS. (c) Représentation en ensembles parallèles (Parallel sets) des survivants du titanic. (d) Représentation en matrice de statistiques sociales sur des pays des Nations Unies, avec l'affichage du dendrogramme correspondant au clustering hiérarchique ayant servi à ordonner les lignes et les colonnes.

Représentation par projection des axes Chaque individu est projeté selon une projection linéaire (ou non-linéaire) des axes dans le plan 2D, comme l'ACP [125] qui est très utilisée en Statistiques par exemple. La projection permet de représenter les données sous la forme d'un nuage de points 2D ou 3D respectant au mieux la structure sous-jacente aux données. La projection peut ensuite être lue comme une métaphore géographique : ce qui est proche est supposé similaire et ce qui est éloigné est supposé dissimilaire. Les projections ont pour principal avantage d'être indépendantes du nombre de dimensions dans les données multidimensionnelles de départ. Elles permettent également de représenter des collections de données [16, 180, 181, 50], où la similarité exprime le contenu des données en les comparant les unes aux autres (Figure 2.3).

Mais la réduction de dimension par des approches non-linéaires ne permet plus de caractériser les données par rapport à leurs valeurs ou leurs dimensions d'origine, ce qui rend impossible certaines tâches d'analyse comme la recherche de tendances, trouver des valeurs maximales, ou détecter des corrélations entre variables. Les interprétations possibles se limitent principalement à l'étude de la structure sous-jacente aux individus et en particulier à l'analyse des relations topologiques entre groupes d'individus similaires (clusters) ou groupes d'individus étiquetés (classes). Au delà de la détection d'outliers et l'énumération des clusters, la projection permet également d'inférer des propriétés sur les données (et indirectement sur la mesure de similarité). Par exemple, on peut vérifier si les clusters sont séparables linéairement dans l'espace des données à l'aide d'une projection linéaire. On distingue différents types de projections de données, principalement les projections linéaires et non linéaires. Différentes approches et techniques de projections sont détaillées plus amplement dans les sections suivantes.



(a) (b)

FIGURE 2.3 – (a) Projection ACP de 128 peintures de Mondrian (1905-1917) [61]. (b) Projection incBoard [173] en grille d'images étiquetées.

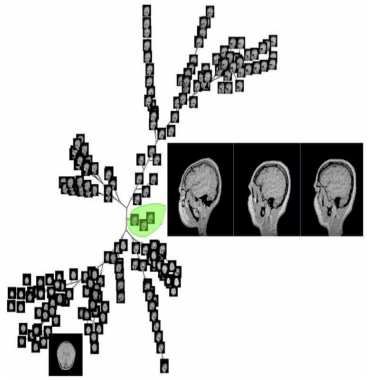
Dans le cadre de l'analyse de corpus de textes, les techniques de projections par réduction de dimension représentent les documents en nuage de points 2D sous la forme d'une métaphore de spatialisation [79, 78, 13], c'est-à-dire d'une représentation en paysage 2D composée d'îles et d'océans ("Information Landscape" [47, 48]) ou bien en paysage 3D composé de collines et vallées ("Themescape" [262]). Les îles ou collines représentent les thèmes des documents du corpus. La taille des îles ou la hauteur des collines permet de représenter la densité de documents et la couleur correspond à d'autres informations comme la date moyenne des documents. Les documents sont positionnés les uns par rapport aux autres selon leurs termes en commun et des océans (ou vallées) séparent les différentes zones de densité, c'est-à-dire des clusters thématiques de documents.

L'utilisation d'un algorithme de clustering hiérarchique aide à définir les différents clusters, comme pour VisIsland [185] qui permet de visualiser sous la forme d'îlots les résultats d'une recherche de documents dans le logiciel xFIND [9]. Ceci permet également d'aider à naviguer dans un espace multi-échelle pour explorer de grandes collections de documents, comme dans le logiciel Infosky [133, 96] qui représente les documents comme les étoiles d'une galaxie séparées par des cellules de Voronoï indiquant les groupes dans la hiérarchie. Le système SPIRE [261] propose une vue en galaxie 2D ainsi qu'une vue 3D en collines et vallées pour analyser des corpus de textes. Le système VxInsight [62, 63, 33] généralise cette visualisation 3D à tout type de collections d'objets abstraits et permet de naviguer dans un espace continue avec différents niveaux d'échelle ainsi que d'effectuer des requêtes pour pouvoir analyser les structures sous-jacentes à de grandes quantités de données. Les Self Organizing Maps (SOM) peuvent aussi être utilisés pour visualiser de larges corpus de texte [111]. Alencar et al. propose un état de l'art de ces techniques de visualisation et analyse de corpus de textes [6]. D'autres applications de la métaphore géographique existent pour créer par exemple une carte de différents groupes de musique [156, 93]. Plus récemment, la métaphore de terrain a également été reprise en visualisation scientifique pour la représentation de données topologiques issues de fonctions scalaires [253, 101, 162].

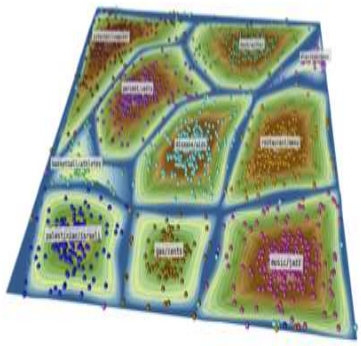
La plupart des représentations par projection utilisent les points pour représenter les individus et certains algorithmes non-linéaires projettent les données en utilisant un positionnement par force (issu des algorithmes de placement de graphes). Par exemple, la technique Radviz [106] positionne chaque dimension le long d'un cercle. Chaque individu est représenté par un point au centre du cercle en fonction du poids de la dimension pour cet individu. Les individus viennent se positionner comme s'ils étaient suspendus au cercle par des ressorts dont la force dépend de leurs valeurs sur les dimensions. L'ordre et la disposition des dimensions change complètement le résultat du placement des points, ce qui peut rendre cette représentation difficile à interpréter.

Une version vectorisée de Radviz [202] découpe chaque dimension en sous-dimensions selon différentes plages de valeurs intéressantes, de manière à pouvoir mieux distinguer l'influence des dimensions dans le placement par force. Cette approche a également été utilisée dans Dust & Magnet [270] où chaque critère d'une dimension vectorisée et positionné interactivement dans l'espace afin que l'utilisateur puisse composer une requête. Les individus sont attirés selon leurs caractéristiques vers les aimants qui représentent les critères et les individus n'ayant aucun critère en commun sont reposés vers les bords de la vue. Cette technique permet ainsi de catégoriser les réponses à une requête selon leurs caractéristiques et peut s'appliquer à de la recherche multi-critères d'information pour que les utilisateurs appréhendent mieux l'impact des différents critères sur la pertinence des réponses.

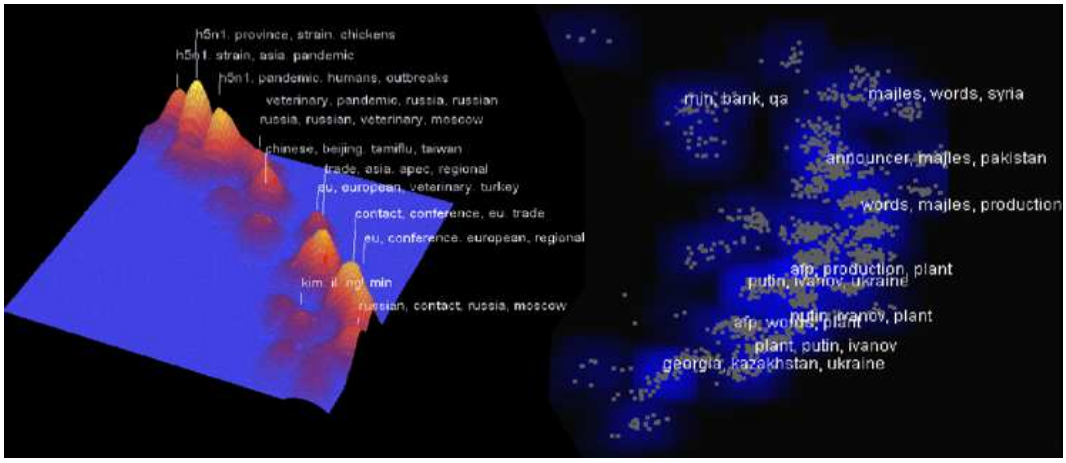
Au delà de l'indépendance du nombre de dimensions qui rend ces techniques viables pour la visualisation de données de grande dimension, les approches par projection ont également pour principal intérêt d'être assez intuitive en exploitant au mieux la variable visuelle de position sur le principe proche ≈ similaire. L'utilisation de métaphores géographiques et d'interactions avec des forces de placement mettent en valeur ce principe. Ces approches peuvent permettre à des non-experts en analyse de données d'appréhender et interpréter des données de grande dimension par le biais de la visualisation (Figure 2.4).



(a) PEx-Image



(b) Topological Landscape



(c) Inspire

FIGURE 2.4 – Représentations par projection des axes. (a) Projection de données d'imagerie médicale [72] (avec NJ-Tree [60]). (b) Projection en paysage topologique 2D de 1896 articles, décrits selon 46393 mots et répartis en 10 catégories, ayant été extrait du New York Times en 2001 [163]. (c) Projection de documents dans le logiciel INSPIRE (représentation Themescape à gauche et en galaxie à droite) [262].

2.1.2 Pipeline de visualisation et interaction

Pour construire une visualisation, les données brutes de départ doivent être transformées selon leur nature et associées à des variables visuelles de manière à obtenir une image des données à partir de laquelle on peut réaliser des tâches précises. Les différentes étapes qui permettent d'aboutir à une représentation efficace des données s'enchaînent selon un pipeline de visualisation. Différentes définitions ont été proposées pour ce pipeline, dont le modèle de référence de la visualisation d'information [45] qui se compose d'une séquence de transformations de données à travers différentes étapes, pour obtenir un rendu final sur l'écran qui peut ensuite être modifié par le biais d'interactions avec chaque étape de transformation (Figure 2.5).

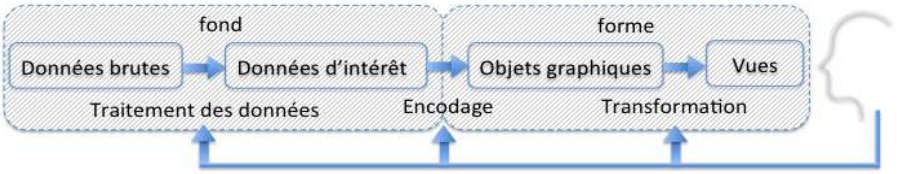


FIGURE 2.5 – Pipeline de la visualisation d'information [45].

De même, le modèle de référence de l'état des données [51] décrit, au travers de différentes étapes et opérateurs, les 4 états de transformation des données abstraites en une représentation (valeur, abstraction analytique, abstraction visuelle, vue). Des opérateurs de transformation changent les données d'un niveau d'abstraction à un autre, alors que des opérateurs d'état modifient les données pour un même niveau d'abstraction. Ces pipelines reprennent une définition plus générale de la visualisation [99] pour l'adapter à la visualisation d'information par la prise en compte de l'interaction ou de niveaux d'abstraction. Le pipeline général se compose de trois étapes :

- filtrage : selon la tâche d'analyse, les données pertinentes sont sélectionnées et enrichies ou réduites.
- association : les données préparées sont associées à des variables visuelles selon leurs propriétés.
- affichage : la géométrie et ses variables visuelles sont rendues à l'écran.

Ce pipeline a été étendu plus tard aux données de grande dimension [70] avec une tâche supplémentaire. Le filtrage est séparé entre l'analyse des données (interpolation, clustering, reconnaissance de formes) et la selection des données d'intérêt.

L'interaction joue également un rôle crucial pour permettre l'exploration des données [51], c'est-à-dire adapter la représentation aux différentes tâches que l'utilisateur est amené à effectuer pour répondre aux questions qui se présentent à lui lors de son analyse des données. On peut classer les interactions selon un certain nombre d'opérateurs d'interactions [269] que nous présentons ci-dessous. Ces opérateurs interviennent dans différents espaces : espace écran (pixels), espace des valeurs (intervalle de filtrage sur une dimension), espace des structures (hiérarchie de données, clustering), espace des variables graphiques (encodage couleur, taille, position). Nous illustrons ces opérateurs par un certain nombre de principes et techniques d'interaction existants, sans être exhaustif, afin de définir le cadre des techniques interactives décrites dans la suite.

Sélectionner Cet opérateur correspond à l'identification d'un objet (ou d'un ensemble d'objets) comme étant jugé intéressant lors de l'exploration de la visualisation ou pertinent par rapport à une tâche. L'élément sélectionné peut ensuite faire l'objet d'opérations comme la mise en valeur, le filtrage, l'ajout à un groupe ou la suppression, le déplacement. Ces opérations s'accompagnent souvent d'une mise en forme de l'objet comme un grossissement des traits de son contour, la mise en transparence, ou l'application d'une couleur correspondant à une étiquette de cluster. Il existe un grand nombre de façons de sélectionner un élément [260]. Sur un nuage de points par exemple, on peut survoler un point avec le curseur de la souris, ou utiliser une brosse (brush) pour peigner des points, ou encore utiliser un lasso pour dessiner les contours d'une surface qui englobe les points à sélectionner. Par exemple, l'affichage excentrique des étiquettes [81] permet de résoudre des problèmes d'étiquetage dans des zones d'occlusions. Cette technique affiche les étiquettes à la périphérie d'une zone de sélection circulaire ou rectangulaire autour du curseur de la souris. Tous les points dans la zone de sélection sont alors reliés par une flèche à leur étiquette en dehors de la zone de sélection.

Le concept de *manipulation directe* [205] est central pour la sélection. Ce principe consiste à manipuler les objets d'intérêt directement sur la visualisation de manière à permettre à l'utilisateur d'effectuer des actions rapides produisant un retour visuel immédiat, comme cela aurait été le cas dans le monde physique où chaque action entraine une réaction. Ce principe permet de rendre les visualisations interactives plus intuitives, de modifier à la demande certains paramètres, ou d'obtenir des informations détaillées sur demande. Ceci rend l'exploration de la visualisation moins frustrante qu'avec une représentation statique.

Explorer ou Naviguer Cet opérateur consiste à changer le point de vue sur les données en choisissant un sous ensemble de données à visualiser, ou en changeant l'orientation de la vue, ou en modifiant le niveau de détail affiché. La navigation est le plus souvent guidée par l'utilisateur dans un but précis visant à répondre à des interrogations lors de la découverte de motifs visuels, ou lorsque des informations pertinentes ne sont pas visibles, ou encore lorsque le niveau de détail n'est pas suffisant. Ce niveau de détail dépend de la quantité d'information à afficher et de sa précision. Pour ne pas perdre l'utilisateur dans une grande quantité d'informations, il est préférable de suivre la mantra de recherche d'information de Shneiderman qui donne un cadre à la démarche de navigation : "Overview first, zoom and filter, details on demand" [206]. L'exploration commence par un aperçu global des données, avant la mise en évidence d'une zone d'intérêt (par zoom et filtrage) dans laquelle on cherche à obtenir des informations détaillées.

La navigation peut également être guidée automatiquement. Par exemple, Grand Tour [12] présente une animation d'une série de projections arbitraires en 2D ou 3D de manière à obtenir différentes perspectives sur les données tout en suivant l'évolution du contexte. Projection Pursuit [88] permet à l'utilisateur de visualiser statiquement différentes projections avec des caractéristiques intéressantes (c'est-à-dire basée sur l'optimisation d'un index d'utilité) mais pouvant être très différentes d'une projection à l'autre. Projection pursuit guided tour [56] combine ces deux techniques pour visualiser des projections pertinentes de manière animée afin de suivre au court du temps le contexte d'une projection à l'autre. L'utilisateur peut interagir afin de visualiser différents pas de temps et observer la transformation de la projection par rapport aux axes.

Reconfigurer Cet opérateur permet d'agir sur l'agencement des données en changeant l'ordonnancement des lignes/colonnes d'une matrice sous forme de *heatmap*, ou en changeant l'ordre des dimensions dans des coordonnées parallèles de manière à mettre en évidence des tendances ou des corrélation entre variables. Pour des données visualisées par projection, différents algorithmes de projection peuvent être utilisés pour choisir celui qui met le mieux en valeur le clustering des données. Dans le cas des projections de données, il existe différentes métriques pour quantifier la qualité visuelle de la projection et ainsi choisir la représentation la plus pertinente.

Différentes solutions d'ordonnancement automatique des lignes et des colonnes de matrices existent [265]. Par exemple dans le cas d'une matrice de similarité, on cherche à réordonner la matrice pour se rapprocher de la matrice optimale dite "Robinsonienne" [179] qui minimise la longueur du chemin liant tous les éléments. Cette matrice optimale permet de mettre en évidence les groupes d'individus similaires dans la matrice, c'est-à-dire le clustering des données. TableLens [176] ou InfoZoom [213] sont des systèmes mêlant visualisation tabulaire et histogrammes des données tout en permettant d'interactivement modifier l'ordre des lignes et des colonnes pour visualiser les données selon différents points de vues pour pouvoir analyser les corrélations entre variables ou différents clustering.

Encoder Cet opérateur permet de modifier soit la technique de visualisation utilisée, soit les variables visuelles utilisées. Différents systèmes de visualisation comme Polaris [215], ayant ouvert la voie ensuite au logiciel Tableau [218], proposent une grande variété de techniques de visualisation pour trouver la *meilleure* représentation selon les données et la tâche considérée. L'encodage des différentes dimensions peut être paramétré à la main par glisser-déposer d'une dimension sur le panneau des variables visuelles. Les échelles de couleur utilisées peuvent également être modifiées selon les préférences des utilisateurs, lesquels sont habitués à des échelles de couleurs spécifiques selon leur domaine d'application. L'encodage couleur doit pouvoir s'adapter aux pratiques liées au domaine d'application et aux éventuelles problèmes de vision des couleurs comme le daltonisme.

L'encodage peut permettre de réduire les problèmes d'occlusion en fonction du clustering des données avec des courbes compactées dans des coordonnées parallèles [273] ou des arêtes compactées dans un graphe [108]. Mais si ces techniques peuvent être activées à la demande de l'utilisateur, tous les encodages ne sont pas équivalents [257] et leur pertinence dépend directement des données et de la tâche considérée. D'où la nécessité d'évaluer de manière quantitative différents encodages possibles d'une même technique, comme les arêtes d'un graphe [109] ou les lignes de coordonnées parallèles [110], pour proposer aux utilisateurs des choix d'encodage efficaces.

Abstraire Cet opérateur permet d'afficher différents niveaux d'abstraction des données. Dans un système de visualisation multi-échelle, la navigation ajuste le niveau de détail en remontant ou en descendant dans la hiérarchie des différentes abstractions de données.Ceci permet à l'utilisateur de naviguer dans les données en passant itérativement d'un aperçu global des données à une vue offrant plus d'information jusqu'à proposer un niveau de détails plus important sur des éléments précis sélectionnés par l'utilisateur. Différentes techniques existent pour visualiser les agrégations hiérarchiques selon plusieurs niveaux d'échelle [75].

Avec les lentilles de déformation comme Magic Lens [24] l'utilisateur peut interactivement déformer une zone de l'écran pour allouer plus d'espace à des éléments l'intéressant. Sur la visualisation, la lentille (Fisheye view) [92, 188, 189, 239] permet d'adapter le niveau de détail dans une région selon le degré d'intérêt des éléments qui la compose. Cette technique de navigation dite "focus+context" permet d'afficher un niveau de détail plus grand dans une zone d'intérêt à l'aide de différentes transformations géométriques [46], tout en préservant le niveau de détail du contexte autour de la zone d'intérêt (une zone de transition fait le lien entre les niveaux de detail).

La navigation "pan+zoom" est également indispensable lorsque l'ensemble des données ne peut pas être affiché dans la fenêtre de visualisation à l'écran (viewport). Elle permet alors de changer la position de la fenêtre dans l'espace visuel de la représentation, avec un zoom géométrique qui grossit ou réduit les éléments graphiques, ou une translation géométrique de la fenêtre à l'aide d'une translation de la souris (pan) ou du déplacement d'une barre de défilement de haut en bas (ou de gauche à droite), de manière à afficher des données initialement non visibles à l'écran.

Filtrer Cet opérateur permet de réaliser des requêtes dynamiques [3, 207] permettant de directement mettre en valeur des données d'intérêt au regard d'une requête de l'utilisateur. La visualisation doit dynamiquement s'adapter aux résultats des requêtes effectuées par l'utilisateur. Par exemple, celui ci peut définir un intervalle de valeur sur une dimension par le biais d'un curseur de défilement (slider). Construire visuellement et itérativement des requêtes de filtrage sur les données permet non seulement d'explorer les données ou d'extraire des informations précises, mais également de palier en partie à certains problèmes dont souffrent les visualisations, comme l'occlusion ou le bruit lié à un trop grand nombre d'individus ou de dimensions.

Par exemple, le système FromDaDy [114] permet de construire des requêtes en brossant le graphe des trajectoires de vols aériens, puis en les déplaçant dans une nouvelle vue pour ainsi itérativement raffiner la recherche. La technique de filtrage croisé introduite [251] et formalisée [252] par Weaver permet de construire une séquence de requêtes à l'aide de sélections et regroupements d'objets entre plusieurs dimensions et jeux de données répartis sur plusieurs vues.

Connecter Cet opérateur affiche dynamiquement les relations entre différentes visualisations ou entre objets. Par exemple dans le système de visualisation de réseaux sociaux Vizster [102], le survol d'un noeud par le curseur peut mettre en valeur ses voisins directs, le double-clic sur un noeud peut afficher des noeuds masqués. Cette approche de "details à la demande" permet de révéler/masquer des éléments précis, comme des métadonnées sur les données brutes, de manière à ne pas surcharger la visualisation qui sert de contexte et de support à la sélection.

Le principe de "link and brush" permet d'afficher les résultats d'une sélection sur différentes visualisations affichées conjointement. On parle alors de vues coordonnées [159], une approche répandue dans la plupart des systèmes de visualisation comme Jigsaw [214], ou XmdvTool [184] (voir l'état de l'art [178] pour plus de détails). En particulier, le système HIVE [182] permet d'associer des blocs (panneaux de paramètres et visualisations) pour construire graphiquement un workflow d'analyse visuelle afin de contrôler dynamiquement un processus de réduction de dimension. Une formalisation de ces systèmes à vues coordonnées a été proposée par Weaver [249] et implémentée dans la toolkit de visualisation Improvise [250].

Dans une matrice de nuages de points, la sélection d'un cluster sur un nuage de points peut mettre en valeur ce cluster dans les autres nuages de points. Cette technique est généralisée dans le système Scatterdice [74] qui permet de naviguer interactivement dans la matrice en affichant le clustering sur le nuage de point central et en montrant comment ce clustering évolue d'un nuage de point à l'autre à l'aide de transitions animées.

Comme nous venons de le voir ces différents opérateurs d'interaction place l'utilisateur au centre de la démarche d'exploration des données afin qu'il puisse répondre à des tâches précises rapidement et dynamiquement. Certaines techniques d'interaction comme les lentilles de déformation cherchent à palier aux problématiques de la visualisation statique d'espaces complexes et d'espaces multi-échelles. Dans cette thèse nous prenons également le parti de palier aux problématiques d'artefacts de projection par le biais de techniques d'interaction et non de mesures représentées statiquement. La section suivante présente les algorithmes de projection permettant de représenter un espace de données de grande dimension.

2.2 Réduction de dimension

Réduire le nombre de dimensions s'avère indispensable dans différents cas d'application comme la classification, la compression ou la visualisation, qui souffrent de problèmes de temps de calcul ou de robustesse lorsque le nombre de dimensions devient trop important dans les données. Différentes approches permettent de réduire le nombre de dimensions tout en veillant à ne pas trop dégrader l'information sous-jacente aux données, comme la sélection de variables (manuelle ou automatique) ou bien l'extraction de caractéristiques [25]. Parmi ces approches de réduction de dimension, ce sont les techniques de projection de données permettant de représenter les données qui nous intéresse ici (Figure 2.6).

En effet, la projection de données permet de visualiser des données de grande dimension sous la forme d'un nuage de points 2D ou 3D tout en préservant la structure sous-jacente aux données. Nous avons vu précédemment que la projection était indépendante du nombre de dimensions contrairement aux autres représentations de données multidimensionnelles. De plus, elle permet également de visualiser les relations de similarité dans des collections d'objets (base d'images, corpus de textes, collection de musiques, ensemble de signaux). L'objectif de la projection est de représenter le plus fidèlement possible la structure sous-jacente aux données afin de détecter des outliers, trouver des clusters et étudier les relations de proximité entre ces différentes structures.

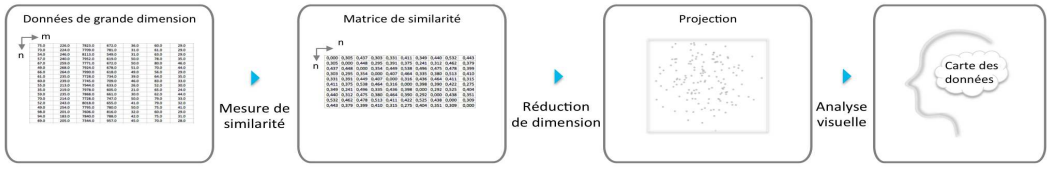


FIGURE 2.6 – Aperçu schématique du processus de visualisation par réduction de dimension.

La section suivante décrit les différentes approches et algorithmes existant pour projeter des données de grande dimension. L'objectif de cette section est d'aider à appréhender les ressorts mathématiques pour mieux comprendre ensuite les enjeux de qualité liés à la projection.

2.2.1 Projection de données

La littérature sur les projections de données est répartie sur différents domaines comme l'apprentissage automatique et la visualisation, dont les thèses respectivement de Venna [242] et de Ingram [116] présentent un état de l'art récent, depuis ses origines en psychologie [271, 138] et en statistiques [170] [232]. Nous proposons ici un aperçu non exhaustif des approches de projection, d'autres états de l'arts peuvent être consultés pour obtenir plus de détails [42, 238, 43, 145, 76]. Nous nous focalisons principalement sur les approches non-supervisée, en citant quelques variantes supervisées c'est-à-dire des variantes utilisant des étiquettes sur les données ou bien nécessitant l'intervention de l'utilisateur pour positionner des points sur la projection.

2.2.1.1 Problème de projection

La projection de données fait l'hypothèse qu'il existe un espace de dimensionnalité inférieure, correspondant idéalement à la dimensionnalité intrinsèque des données, dans lequel on peut plonger les données. La dimensionnalité intrinsèque des données peut être vue comme le minimum de dimensions requises pour exprimer toutes les propriétés sur les données [91]. On peut catégoriser les algorithmes de projection en deux familles : linéaire et non-linéaire.

Les algorithmes de projection linéaires font l'hypothèse que l'on peut trouver un espace de plus faible dimension, obtenu par combinaison linéaire des dimensions de l'espace de départ, dans lequel projeter orthogonalement les données. Mais cette hypothèse n'est pas toujours vérifiée sur des données réelles, dans lesquelles les relations entre données sont souvent non-linéaires. La structure intrinsèque des données peut avoir une géométrie complexe, c'est-à-dire être associée à une variété topologique avec des courbures qu'une projection orthogonale ne peut pas respecter. Les algorithmes de projection non-linéaires font l'hypothèse qu'il existe une variété non-linéaire (manifold) de plus faible dimension, capturant la complexité intrinsèque des données, que l'on peut projeter sans trop de déformation dans un espace à deux ou trois dimensions.

Le problème de projection de données peut s'écrire formellement comme suit :
Pour un ensemble X de n points dans l'espace des données m-dimensionel $X \in \mathbb{R}^{n\times m}$ devant être projeté dans un espace de projection Y à p dimensions $Y \in \mathbb{R}^{n\times p}$, considérant une métrique de distance (ou de dissimilarité) sur l'espace des données $d_m : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ et sur l'espace de projection $d_p : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, l'application de projection ψ s'écrit :

$$\psi : \begin{array}{c} \mathbb{R}^m \to \mathbb{R}^p \\ x_i \mapsto y_i \end{array} \quad \text{tel que } d_m(x_i,x_j) \approx d_p(y_i,y_j), \text{ pour tout } (x_i,x_j) \in X \text{ et } (y_i,y_j) \in Y$$

Du faite de la réduction de dimension, l'application ψ introduit une erreur, appelée stress, dans l'approximation des paires de distances. Les différentes techniques de projection cherchent à minimiser cette erreur en ayant recours à différents critères et approches d'optimisation. Le stress ε_ψ,

dans sa définition originale [228], s'écrit comme la somme quadratique des écarts de distances :

$$\varepsilon_{\psi} = \sum_{i,j}^{n} [d_m(x_i, x_j) - d_p(y_i, y_j)]^2 \tag{2.1}$$

2.2.1.2 Mesure de similarité

L'application de projection repose sur la définition d'une métrique de distance dans l'espace des données et d'une autre métrique dans l'espace de projection. La projection vers un espace en trois dimensions est fréquemment utilisée en analyse de données, mais elle pose des problèmes d'occlusion et d'interaction pour naviguer dans l'espace de projection. Aussi dans cette thèse, nous ne considérons que des projections vers un espace en deux dimensions (2D) pour visualiser les données, car la projection en 2D est "suffisamment efficace" [197] pour séparer les clusters. Dans un espace 2D, la distance Euclidienne, $||\overrightarrow{uv}|| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2}$, avec $(u, v) \in \mathbb{R}^2$, est la plus intuitive du point de vue de la perception humaine, aussi nous ne considérerons dans la suite que cette métrique de distance dans l'espace de projection.

Dans l'espace des données, différentes mesures sont possibles pour quantifier la similarité entre deux objets. Cette mesure dépend fortement de la sémantique sous-jacente à la notion de similarité, laquelle est directement liée au domaine d'application ainsi qu'à la nature des données (table numérique, ensemble d'images, corpus de textes, signaux multi-capteurs, etc.). Par exemple, dans une collection de portraits photos, on peut aussi bien juger deux photos comme similaires parce qu'elles ont les mêmes caractéristiques de couleurs et textures (similarité basée sur l'extraction de caractéristiques) ou bien parce qu'elles représentent deux personnes qui se ressemblent (similarité basée sur la reconnaissance de motifs).

La représentation des données doit permettre d'extraire facilement la relation de similarité entre chaque paire d'individus, c'est-à-dire la matrice de similarité définissant l'espace des données. C'est pourquoi la projection de données repose sur un encodage intuitif des similarités par le biais de la variable visuelle de position. La projection repose ainsi sur une métaphore géographique qui met en correspondance deux grandeurs : la similarité entre deux objets au regard de leurs dimensions et la distance entre ces mêmes objets dans un plan 2D. Plus les objets sont similaires, plus ils devront être proches sur la projection, c'est-à-dire une grande similarité (ou faible dissimilarité) correspond à une grande proximité (ou faible distance) sur la projection. On peut préférer mesurer la dissimilarité plutôt que la similarité, car une grande distance sur la projection correspond à l'inverse d'une grande similarité, c'est-à-dire une grande dissimilarité. Dans la suite, on appellera *proximités* les relations de similarité entre individus dans l'espace des données.

Une dissimilarité métrique (ou pseudo métrique) dans l'espace des données doit correspondre à la définition mathématique d'une distance $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ telle que, pour tout $(u, v, w) \in \mathbb{R}^{3\times m}$, la distance d satisfait les conditions suivantes :

- définition ($d(u, v) = 0 \Leftrightarrow u = v$)
- positivité ($d(u, v) \geq 0$)
- symétrie ($d(u, v) = d(v, u)$)
- inégalité triangulaire ($d(u, v) \leq d(u, w) + d(w, v)$)

La plupart des métriques usuelles, comme la distance de Manhattan $d(u,v) = \sum_{i=1}^{n} |u_i - v_i|$ ou la distance Euclidienne $d(u,v) = \sqrt{\sum_{i=1}^{n} |u_i - v_i|^2}$, sont des cas particuliers de la distance de Minkowski $d(u,v) = (\sum_{i=1}^{n} |u_i - v_i|^p)^{\frac{1}{p}}$. Ces mesures supposent une indépendance entre dimensions mais elles peuvent souffrir de la domination d'une dimension qui aurait des valeurs réparties sur un plus large spectre que les autres. La normalisation des dimensions ou la pondération permet de résoudre les problèmes de domination, mais il faut prendre en compte ces poids dans l'interprétation de la similarité, de même que les corrélations entre variables peuvent déformer la mesure. La distance de Mahalanobis $d(u,v) = \sqrt{((u-v)^T S^{-1}(u-v))}$, avec S la matrice de covariance entre variables, prend en compte les corrélations entre variables pour les pondérer. On notera que si la matrice de covariance est diagonale, c'est-à-dire s'il y a indépendance des dimensions, on obtient la distance Euclidienne normalisée $d(u,v) = \sqrt{(\sum_{i=1}^{p} \frac{(u_i - v_i)^2}{\sigma_i^2})}$, avec σ_i l'écart type de u_i.

On considère dans cette thèse des données numériques et des collections d'objets. On ne considère pas des données avec des relations binaires ou ordinales qui ont leurs propres métriques. En analyse de textes [187], le cosinus de l'angle entre les vecteurs de mots clés est souvent utilisé pour mesurer la ressemblance entre textes. En analyse d'images [95], la distance Euclidienne est couramment utilisée pour mesurer des distances entre pixels ou entre caractéristiques extraites automatiquement. Pour comparer l'expression de gènes, on utilise souvent la corrélation de Pearson qui est également non métrique. Il existe également des dissimilarités non-métriques, aussi appelées relations explicites [68], qui dépendent d'une valuation manuelle des similarités. Ce type de dissimilarités est fréquent en Psychométrie mais il est également utilise pour comparer par exemple des groupes de musique entre eux [93].

La sémantique sous-jacente à la similarité est importante, car elle permet ensuite d'interpréter le sens des différents motifs observés sur la projection. Les distances métriques amalgament les différentes variables dans les données, aussi la tâche d'interprétation à posteriori des caractéristiques des clusters observés sur la projection n'est pas toujours évidente. Dans cette thèse, on ne s'intéresse pas à définir ou choisir les caractéristiques permettant de construire une métrique de distance, mais nous considérons une mesure de distance, fournie par un expert ou standard par rapport au cas d'application, permettant de définir des structures dans les données reflétant au mieux la réalité. Une fois cette mesure définie sur l'espace des données, il existe un large panel d'algorithmes implémentant l'application de projection ψ.

Dans cette section, nous présentons les approches linéaires puis non-linéaires, en introduisant les différentes approches mathématiques utilisées afin de mieux comprendre dans la suite les enjeux de mesure de la de qualité associée aux projections.

2.2.1.3 Approches linéaires

L'Analyse en Composantes Principales (ACP) [170] est une projection linéaire qui cherche à préserver au mieux la variance dans les données. Du point de vue des statistiques, elle transforme des variables liées entre elles, dites "corrélées", en de nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées "composantes principales", ou axes principaux indépendants et expliquent la variabilité dans les données. L'ACP capture donc l'inertie maximale des données pour réduire le nombre de variables et rendre l'information moins redondante.

Du point de vue géométrique, l'ACP est une projection orthogonale des données dans un sous espace principal obtenu par combinaison linéaire des axes de départ (c'est-à-dire les dimensions dans les données). Le sous espace est calculé de telle sorte que le nuage de points résultant de la projection maximise la variance dans les données. En termes mathématiques, l'ACP de données X cherche une combinaison linéaire M qui maximise $M^t cov(X)M$, avec $M^t M = 1$ et cov(X) la matrice de covariance (ou la matrice de corrélation si les données sont centrées réduites). Il peut être montré que la combinaison linéaire optimale se compose des deux vecteurs propres (ou composantes principales) de la matrice de covariance ayant les plus grandes valeurs propres (ou des trois premiers vecteurs propres si la projection est visualisée en 3D). L'algorithme repose donc sur une décomposition en valeurs propres λ et vecteurs propres de la matrice de covariance pour résoudre le problème : $cov(X)M = \lambda M$. Les données sont ensuite projetées orthogonalement dans le repère des deux premières composantes principales pour être visualisée en 2D (Figure 2.7).

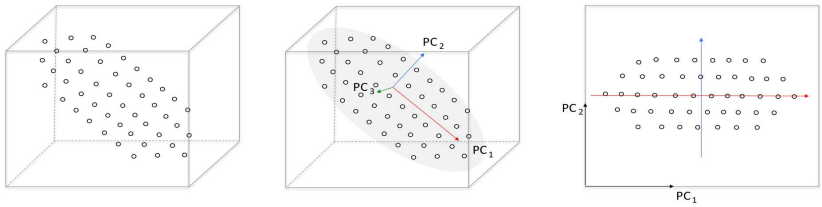


FIGURE 2.7 – Schéma illustrant la projection en 2D, par Analyse en Composantes Principales, d'un nuage de points en 3D.

Cependant si cette technique capture aisément des clusters séparables linéairement dans les données, elle ne permet pas de séparer des relations non-linéaires et introduit alors des distorsions de faux voisinages. Mais sans information sur ces distorsions, on ne peut donc pas inférer à partir de la projection que les données sont séparables linéairement. L'ACP est néanmoins très utilisées dans de nombreux domaines d'applications pour ses bonnes propriétés. En effet, contrairement au autres techniques de projection, l'ACP est réversible et elle fournie une pondération des composantes principales permettant ainsi de déterminer les directions de dispersion des données les plus importantes. L'ACP donne ainsi des informations sur la dimensionnalité intrinsèque des données à partir des écarts dans le spectre des valeurs propres. De plus, les vecteurs propres décrivent la contribution de chaque variable de départ à la projection, ce qui rend la projection directement interprétable. De nombreuses variantes de l'ACP [125] existent pour accélérer son calcul ou améliorer sa robustesse notamment aux outliers [112].

Le Linear Multidimensional Scaling (MDS) [228], aussi connu sous le nom de "Classical MDS", projette également les données dans un sous-espace linéaire. Cette approche cherche à préserver au mieux dans l'espace de projection les paires de distances au carré issues de l'espace des données. Pour cela, le Linear MDS trouve une combinaison linéaire optimale pour toute métrique définie dans l'espace des données. L'intuition sous-jacente est semblable à celle de l'ACP et les points sont projetés dans un sous espace linéaire de variance maximum. La différence vient de la décomposition en vecteurs propres de la matrice de Gram, c'est-à-dire la matrice des produits scalaires entre vecteurs d'individus et non celle des covariances. Cette technique souffre ainsi des mêmes problèmes que l'ACP pour projeter des données non-séparables linéairement. Contraire-

ment à l'ACP et comme toutes les autres techniques décrites dans la suite, cette technique prend directement en entrée une matrice de similarité, c'est pourquoi les axes de la projection de ces techniques ne sont pas interprétables.

Toutes les techniques de projection qui se basent sur une décomposition en vecteurs et en valeurs propres sont appelées "approches spectrales". Il a été montré [136] qu'en maximisant la variance, l'ACP minimise également le stress de la projection, c'est-à-dire la somme quadratique des écarts entre les paires de distances Eucidiennes dans les espaces de projection et de données. C'est pourquoi le Linear MDS est équivalent à l'ACP avec une distance Euclidienne au carré. Il est a noter que le temps de calcul de ces méthodes dépendent du nombre de points et non du nombre de dimensions et pour des jeux de données trop grand, différentes approximations de la décompositions en valeurs singulières peuvent être utilisées pour accélérer le calcul [174].

2.2.1.4 Approches non-linéaires

De nombreuses techniques de projection non-linéaires ont été développées pour projeter les données tout en préservant les propriétés de leur variété sous-jacente [94]. Ces techniques optimisent différentes variations de la mesure de stress basées, soit sur les distances entre données (on parle de MDS métrique), soit sur les rangs des distances (on parle de MDS non-métrique). Elle utilisent ensuite différentes méthodes d'optimisation (décomposition spectrale, descente de gradient, placement par forces, réseau de neurones, méthode à noyau) afin de permettre de trouver un optimum global lorsqu'elles préservent la structure globale, ou bien un optimum local lorsqu'elles préservent les voisinages locaux.

Approches basées sur les distances globales

Il existe différentes variantes du MDS traditionnel [28], mais elles ont toutes pour point commun de chercher à trouver une projection qui préserve au mieux les paires de distances d'origine. Ainsi la formulation du stress brute [138] ε_ψ (2.1) évolue d'une technique à l'autre, pouvant devenir probabiliste [275], ou donner plus d'importance à certains critères. Par exemple, la projection non-linéaire de Sammon [186] préserve les faibles distances normalisées par les distances d'origine :

$$\varepsilon_\psi = \sum_{i,j}^{n} \frac{(d_m(x_i,x_j) - d_p(y_i,y_j))^2}{d_m(x_i,x_j)}$$

L'Analyse en Composantes Curvilignes (CCA) [66] est une variante du MDS traditionnel qui ne préserve pas toutes les distances mais seulement celles des points proches sur la projection. Elle optimise cette fonction objectif :

$$\varepsilon_\psi = \frac{1}{2}\sum_{i}^{n}\sum_{i\neq j}^{n}[d_m(x_i,x_j) - d_p(y_i,y_j)]^2 F_\sigma(d_p(y_i,y_j))$$

Avec F_σ une fonction de Heaviside : $F_\sigma(u) = 1, \text{si } u \leq \sigma$ et $F_\sigma(u) = 0$ sinon. L'optimisation est réalisée avec une descente de gradient stochastique qui réduit progressivement la valeur de σ de manière à préserver au fur et à mesurer les structures locales dans l'espace de projection contrairement à la projection de Sammon.

Le MDS non-métrique de Kruskal [138] se base sur une optimisation du stress brut en 2 temps : d'abord une transformation monotone optimale des dissimilarités en distances métriques qui préservent le rang des dissimilarités, puis une optimisation des distances de rang itérativement pour équilibrer le stress et la monotonie. Pour minimiser le stress, une descente de gradient basée sur la méthode d'Euler [11] permet d'approximer une solution à partir d'une initialisation aléatoire. La convergence vers un optimum local peut être accélérée avec d'autres méthodes comme celle de Newton [127] ou SMACOF [64].

L'approche de Chalmers [48], reprise dans sa version parallèle avec Glimmer [118], utilise un algorithme de placement de graphe par force pour minimiser le stress brut et converger très rapidement pour des jeux de données de très grande taille. L'approche hiérarchique de Glimmer permet d'éviter de tomber dans un optimum local mais les aspects aléatoires du placement par force ajoute tout de même un peu de bruit à la projection. L'approche hiérarchique permet de visualiser des jeux de données de très grande taille, comme MDS Steer [259] qui affine à la demande le nombre de points projetés et la précision de la projection.

D'autres techniques dérivée du Classical MDS, prévilégient une résolution algébrique (par décomposition spectrale) plutôt qu'un processus d'optimisation. La Kernel PCA [191] est une variante de l'ACP et du Classical MDS capable de décrire des données non-linéaires en ce basant sur deux hypothèses : il existe un espace des caractéristiques dans lequel les données sont linéaires et il existe une application qui approxime le produit scalaire dans cet espace. Cette application est appelée *noyau* et l'utilisation d'un noyau non-linéaire est appelé le "kernel trick". Les coordonnées dans cet espace restent inconnues, mais le noyau permet d'utiliser une matrice de Gram dans l'espace des caractéristiques. La décomposition en valeurs et vecteurs propres de cette matrice de Gram des produits scalaire capture les relations non-linéaires en maximisant la variance dans l'espace des caractéristiques. La difficulté de cette approche est d'utiliser le "bon" noyau, ce qui nécessite une connaissance a priori des données ou bien un apprentissage non supervisé des relations de proximité entre les données. D'autres variantes du MDS traditionnel propose une approche basée sur la préservation des structures de la variété, par le biais notamment des distances géodésiques. On parle d'apprentissage de la variété (manifold learning) car la définition de ces distances entre voisins revient à approximer la variété.

Approches basée sur l'apprentissage de la variété

Isomap [226] modélise les proximités en termes de distance géodésique plutôt que de chercher à les optimiser directement sur la projection. Cette variante du Classical MDS utilise des distances métriques géodésiques qui sont apprises en approximant linéairement la variété. Un graphe non orienté des k-plus proches voisins est construit en utilisant pour chaque noeud leurs k plus petites dissimilarités afin de définir leurs arrêtes pondérées. Ces poids représentent l'approximation locale de la distance géodésique sur la variété. Une matrice des distances géodésiques est dérivée de ce graphe, par un algorithme de plus proche chemin, pour ensuite être décomposée en composantes principales servant à projeter orthogonalement les données. Des surestimations des vraies distances peuvent générer des trous, la prise en compte de la densité des données dans Fast Isomap [210] résout ce problème. La technique Maximum Variance Unfolding (MVU) [254] conserve aussi les distances géodésiques mais diffère d'ISOMAP dans sa maximisation des distances entre points projetés de manière à déplier au maximum la variété.

LLE [183] modélise également la variété en extrayant sa géométrie intrinsèque par le biais d'un graphe des k-plus proches voisins. Cependant à l'inverse d'ISOMAP qui utilise une approche globale, LLE repose sur une approche locale qui vise à préserver la géométrie locale en approximant par combinaison linéaire convexe les k-plus proches voisins de chaque donnée. On suppose que pour une donnée considérée, ses k-plus proches voisins forment un plan local unique dont elle est le centre, on peut alors représenter chaque point par une combinaison linéaire de ses k-plus proches voisins. La matrice W des poids de reconstruction issue de ces combinaisons linéaires est obtenue par minimisation du problème linéaire de fonction objectif : $\psi(W) = \sum_{i=1}^{n} \frac{|x_i - \sum_{j=1}^{k} w_{i,j} x_i^{(j)}|^2}{d_m(x_i, x_j)}$, avec $\sum_{j=1}^{k} w_{i,j} = 1$ et $x_i^{(j)}$ le j-ème voisin du point x_i. Comme la géométrie locale intrinsèque a pour propriété d'être insensible aux transformations géométriques, la matrice W des poids de reconstruction est préservée après projection. On peut ainsi montrer que l'on peut dériver une projection orthogonale des données qui minimise la fonction objectif à partir des composantes principales issues de la décomposition en vecteurs propres de la matrice des poids de reconstruction $(I - W^t)(I - W)$. LLE n'est pas toujours plus efficace qu'ISOMAP, car il est sensible aux variétés composées de trous [190] et a tendance à replier un grand nombre de données sur un unique point en 2D.

Laplacian Eigenmap [18] est une technique proche de LLE qui préserve la structure locale. Cette technique préserve les paires de distances entre k-plus proches voisins. Elle construit un graphe des k-plus proches voisins où le poids sur chaque arrête correspond à une fonction de noyau Gaussien : $w_{i,j} = \exp -\frac{|x_i - x_j|^2}{2\sigma^2}$ avec x_j voisin de x_i et $w_{i,j} = 0$ sinon. La matrice Laplacienne L correspond à la différence entre la matrice diagonale des rangs M ($mii = \sum_j w_{i,j}$) et la matrice d'adjacence du graphe W : $L = M - W$. Il peut être montré que la projection optimale qui minimise la fonction objectif $\psi(Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} (|y_i - y_j|^2 w_{i,j}) = 2Y^t LY$ peut s'obtenir par projection orthogonale sur les composantes principales issues de la décomposition en vecteurs propres de la matrice Laplacienne. La fonction de noyau Gaussien met en valeur les faibles distances plutôt que les grandes distances, donc les voisins les plus proches sont ceux qui contribuent le plus.

Au delà des variantes de ces techniques, différentes approches non-spectrales utilisent les distances géodésique comme l'Analyse en Distances Curvilignes (CDA) [142] qui utilise le même stress que CCA. Une autre variante de CCA utilise un placement par force, DD-HDS [149]. Cette technique propose également une pondération des distances prenant en compte le phénomène de concentration de la mesure [1] qui constate que dans les espaces de grande dimension les distances entre données sont très semblables et convergent vers une même moyenne. La technique RankVisu [148] prend également en compte ce phénomène mais en optimisant une fonction objectif basée sur les distances géodésiques. Ces différentes méthodes, dont celles utilisant les k-plus proches voisins, impliquent souvent beaucoup de paramètres rendant complexe la configuration.

Différentes variantes basées sur des points de contrôles existent également pour réduire les temps de calcul des méthodes spectrales comme FastMap [80, 225], Piecewise Laplacian-based Projection (PLP) [169, 165], ou comme Pivot MDS [34] ou Landmark MDS [65] qui nécessitent que l'utilisateur positionne manuellement ces points. Différentes projections itératives et interactives comme Local Affine MDS (LAMP)[124] ou iLAMP [175] ont ainsi été développées pour aider à mieux appréhender les paramètres de la projection. Il existe également des approches de projection supervisée comme l'analyse discriminante linéaire qui sépare linéairement des clusters prédéfinies ou des méthodes non-linéaires qui cherchent à séparer les classes [67] ou à optimiser

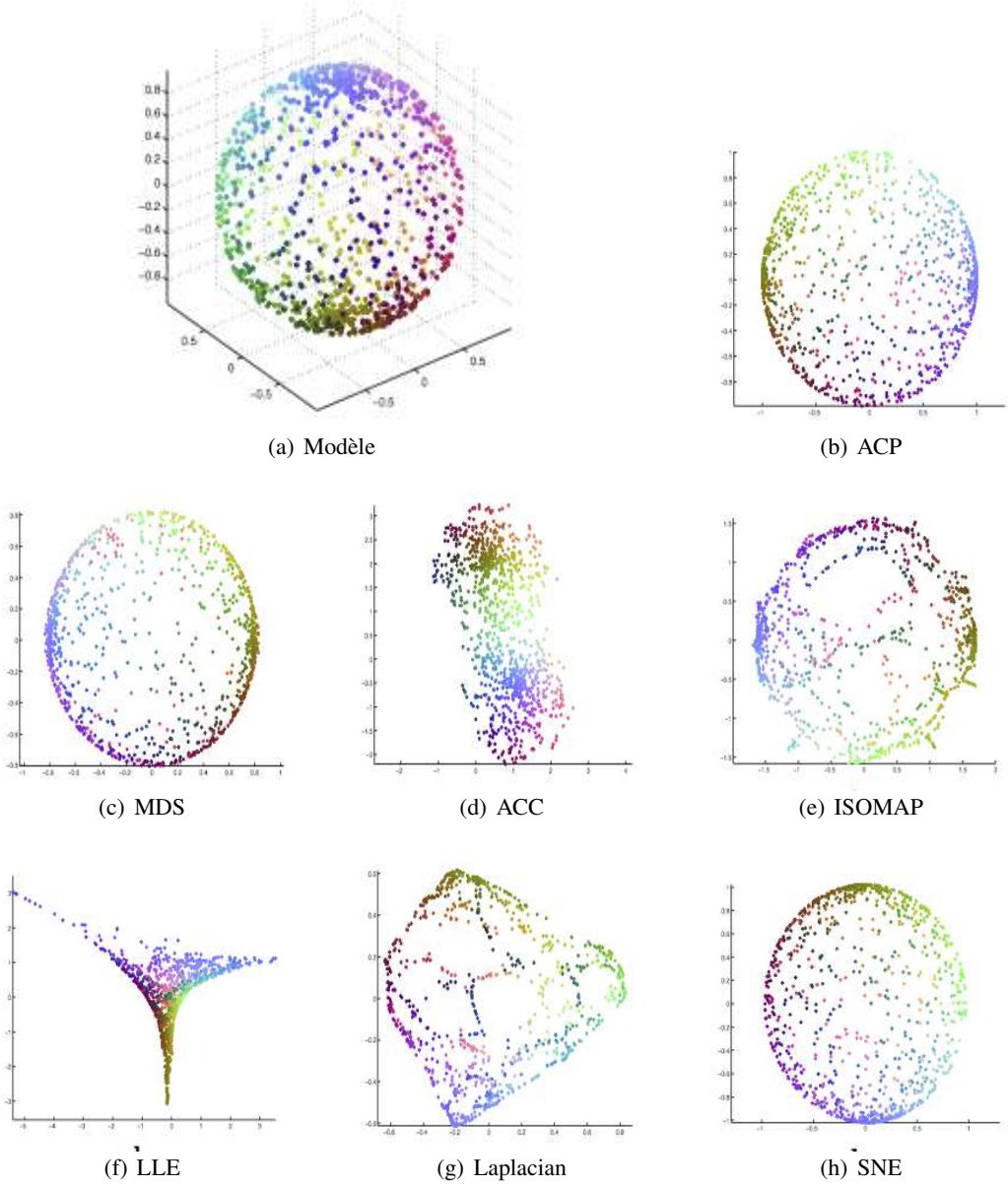
le placement des clusters en fonction des classes comme ClassiMap [147]. D'autres techniques utilisent des projections sous contrainte comme la projection en grille hexagonale [172] ou rectangulaire [173]. La projection en arbre phylogénétique (NJ Tree) [60] par jointure de voisins permet également d'obtenir une projection atypique qui met en évidence la densité des clusters en fonction de la taille des branches de l'arbre.

Approches probabilistes ou stochastiques

Les projections probabilistes comme SNE [104] ou t-SNE [237] minimisent la divergence de Kullback-Leibler [139] entre les probabilités dérivées des distances dans l'espace des données et dans l'espace de projection. Dans SNE, des distributions conditionnelles de probabilité sont construites pour chaque point à partir des distances dans l'espace des données, où la probabilité est interprétée comme la vraisemblance qu'une donnée soit le j-ème plus proche voisin d'une autre donnée. On peut définir la distribution $p_{i,j}$ dans l'espace des données et la distribution $q_{i,j}$ dans l'espace projection : $p_{i,j} = \frac{exp(-d(x_i,x_j)/\sigma_i^{(in)})}{\sum_{k\neq i} exp(-d(x_i,x_k)/\sigma_i^{(in)})}$, $q_{i,j} = \frac{exp(-||y_i-y_j||^2/\sigma_i^{(out)})}{\sum_{k\neq i} exp(-||y_i-y_k||^2/\sigma_i^{(out)})}$, avec $\sigma_i^{(in)}$ le paramètre qui contrôle la taille de la Gaussienne. Ce paramètre peut être fixé manuellement ou bien en fonction de l'entropie de la distribution. Intuitivement ces méthodes assignent une plus grande probabilité aux données proches entre elles. Si $\sigma_i^{(out)} = 1$, alors la projection préserve les probabilités et si $\sigma_i^{(out)} = \sigma_i^{(in)}$ alors elle préserve les distances. La projection qui minimise la divergence de Kullback-Leibler satisfait donc la fonction objectif : $\psi(Y) = \sum_i \sum_j p_{i,j} log \frac{p_{i,j}}{q_{i,j}}$. Une autre approche comme NeRV [245, 243] optimise des mesures de probabilité basées sur celles de Local MDS [49] qui utilise un critère de continuité locale basé sur des mesures de précision/rappel. Un paramètre permet d'équilibrer la fonction objectif pour préserver plutôt les densités de probabilités dans l'espace des données ou bien dans l'espace de la projection. L'intérêt de ce type de projection est de séparer relativement bien les clusters entre eux et d'éviter l'effet de concentration des points en un seul "blob" central.

Approches par carte auto-organisée

Les cartes auto-organisées (Self Organizing Map), introduites par Kohonen [135], projettent les données en les discrétisant sur une grille (hexagonale ou rectangulaire) par apprentissage d'un réseau de neurones, tout en préservant les propriétés topologiques. Chaque case de la grille correspond à un neurone auquel est associé un vecteur de poids dans l'espace de grande dimension. Les données sont placées sur la carte en trouvant le neurone avec le vecteur poids le plus proche. A chaque itération, la fonction objectif suivante est optimisée par tirage aléatoire d'une donnée x : $c(t) = argmin_j\{d(x(t),m_j)\}$, avec $d(x(t),m_j)$ la distance entre l'individu x sélectionné à l'itération t et le neurone (ou prototype) m_j. Avant de passer à l'itération suivante, une fois le neurone optimum m_j trouvé, sa position est mise à jour : $m_j(t+1) = m_j(t) + h_{c(t),j}(t)[x(t) - m_j(t)]$, avec $h_{c(t),j}(t) = h(||r_c(t) - r_j||;t)$ la fonction de voisinage (une fonction Gaussienne peut par exemple être utilisée) et r_j et $r_c(t)$ les positions sur la grille. Les SOM sont calculées en batch ou en séquentiel et il existe un grand nombre de variantes selon la fonction objectif, la règle de mise à jour ou la topologie. Nous ne rentrons pas plus dans le détail des SOM dans la suite de cette thèse, car nous considérons uniquement des projections où chaque individu est représenté directement sur la visualisation et non synthétisé par un neurone, c'est-à-dire une quantification vectorielle.



(a) Modèle (b) ACP

(c) MDS (d) ACC (e) ISOMAP

(f) LLE (g) Laplacian (h) SNE

FIGURE 2.8 – Différentes projections d'une sphère 3D [242]. On remarque que chaque algorithme produit un nuage de points avec des caractéristiques spécifiques. Par exemple, LLE produit une projection difficilement exploitable. L'ACC en revanche à dépliée la sphère en la découpant en deux hémisphères qu'elles a ensuite projeté. Cette projection présente donc des distorsions de type déchirure, là où les autres projections introduisent plutôt des erreurs locales de voisinages.

2.2.2 Qualité de la réduction de dimension

Il existe donc une grande diversité d'algorithmes de projection et chaque technique implique différentes hypothèses, critères et paramètres, qui sont susceptibles d'influencer le nuage de points résultant, comme par exemple le choix d'une hypothèse de linéarité ou d'un critère d'approximation de la variété qui peut être local ou global et avec une méthode spectrale ou bien par optimisation. Aussi la projection d'un même jeu de données peut changer du tout au tout (Figure 2.8 et Figure 2.9). Le choix d'une approche adaptée dépend de nombreux facteurs liés à la nature intrinsèque des données, comme la présence d'une topologie particulière ou des variations de densité importantes entre les régions de l'espace des données ; autant de caractéristiques qui peuvent influencer plus ou moins fortement chaque algorithme de projection. Ces facteurs sont difficiles à déterminer et requièrent une expérience pointue dans le domaine des projections de données.

Pour autant, il possible de quantifier après projection la qualité de la projection selon la préservation de l'information d'origine ou les caractéristiques visuelles du nuage de points. Différentes mesures de qualité existent afin d'aider des utilisateurs non-experts en projections à appréhender ces enjeux de qualité pour qu'ils puissent explorer différents choix d'algorithmes et de paramétrage dans le but de sélectionner de "bonnes" projections. La qualité peut également être utilisée dans un processus automatique pour proposer de "bonnes" projections à l'utilisateur sans qu'il n'ait à se soucier de leur configuration [23]. La visualisation de la qualité locale de la projection permet également d'essayer de mieux comprendre les particularités de la projection afin de mieux appréhender la structure des données ayant pu être altérée par la réduction de dimension.

Dans cette section, nous présentons les différentes mesures existantes dans la littérature pour déterminer la qualité de la réduction de dimension [83], au travers de différentes approches, basées soit sur la qualité de la préservation des structures d'origine, soit sur la qualité visuelle de du nuage de points.

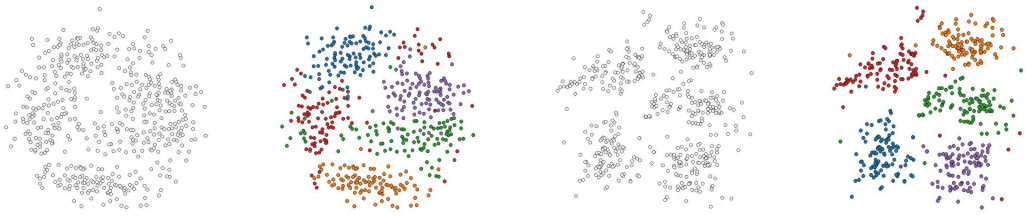


FIGURE 2.9 – Projection Classical MDS (a,b) et NeRV (c,d) d'un échantillon du jeu de données Optical Recognition of Handwritten Digits ([87]) : 32x32 bitmaps normalisés de chiffres écrits à la main (0-9) ont été extraits à partir de chiffres écrits à la main par 43 personnes. Les bitmaps ont été ensuite divisés en blocs non chevauchant pour générer une matrice 8x8 où chaque élément est un entier dans un intervalle de 0..15. Nous avons utilisé un échantillon composé de 5 chiffres impairs (1, 3, 5, 7, 9) avec 100 individus choisis aléatoirement pour chaque chiffre. On remarque que la projection NeRV semble séparer mieux les clusters ainsi que les classes, mais on ne peut pas être sûr que ces projections représentent fidèlement les structures d'origine.

Qualité de la projection

Les mesures de qualité liées au stress (2.1) de la projection vérifient que la projection 2D a préservé localement les proximités de l'espace des données dans l'espace de projection. Plus les proximités 2D correspondent aux proximités d'origine et plus le stress est faible. On pourrait donc prendre la dernière valeur de la fonction objectif optimisée par l'algorithme de projection pour obtenir une idée de la qualité de la projection. Cependant, comme nous l'avons vu précédemment, chaque technique de projection optimise une fonction objectif qui lui est propre, avec différents paramètres, ce qui rend inéquitable la comparaison entre les valeurs des fonctions objectifs tel quel. Il faut donc des mesures indépendantes de la technique de projection utilisée.

Différentes mesures existent dans le domaine de l'apprentissage, ces mesures s'inspirent des fonctions objectifs développées pour réduire la dimension mais sans les contraintes de continuité ou différentiabilité requises par ces algorithmes. La plupart de ces mesures sont globales et utilisent les rangs des proximités comme la mesure de précision/rappel [49] ou de fiabilité/continuité [244]. Ces mesures basées sur les rangs ont récemment été unifiées au sein d'une unique mesure de qualité dépendant d'un seul paramètre résumant la taille du voisinage et le facteur d'échelle [143, 144]. Ces mesures sont pratiques pour comparer automatiquement des projections entre elles mais elles n'expliquent pas d'où proviennent les pertes de qualité.

Visualisation du Stress

Dans le domaine de la visualisation d'information, les mesures de qualité sont plus locales afin d'être visualisées en chaque point et d'aider à expliquer la projection. Kruskal [138] propose de comparer visuellement les proximités en utilisant un *diagramme de Shepard*. Les proximités dans l'espace 2D sont affichées en comparaison des proximités dans l'espace des données. La proximité des points à la droite $y = x$ mesure la qualité de la projection en termes de préservation des proximités. Cependant, cette technique requiert de la pratique car cela implique de visuellement lier deux vues différentes où les points ont un sens différent : la visualisation de la projection où les points représentent des individus et le diagramme où les points représentent des paires d'individus.

D'autres méthodes affichent la qualité directement sur la projection, en utilisant un "jitter disc" autour de chaque point dont le rayon indique le stress [37] et un deuxième cercle indique par son rayon la variation du stress au cours de l'optimisation. Cette technique souffre cependant de problème de chevauchement entre les cercles, aussi une échelle de couleur est plus fréquemment utilisée pour représenter le stress. Un encodage courant consiste à colorer chaque point de la projection en fonction de leur valeur du stress local par le biais d'une échelle de couleur relativement intuitive du jaune (stress faible) au rouge (stress élevé) [21]. Ainsi la couleur rouge indique les points les moins fiables de la projection, c'est-à-dire ceux dont le voisinage d'origine est le moins respecté et donc ceux pour lesquels le risque de faire des erreurs d'interprétation est le plus élevé. Seifert [199] propose de visualiser une mesure de stress local en arrière plan *d'un paysage d'information*, en interpolant la couleur, pour la visualisation de corpus de texte. Schreck [194] propose également une autre mesure de stress local, nommée mesure de précision. Cette mesure est visualisée également en utilisant différents types d'interpolation de la couleur entre les points, selon différents paramètres, pour former une *carte de précision*. Toutefois ces techniques n'indiquent pas où les points fortement stressés, c'est-à-dire mals placés sur la projection, devraient en réalité être positionnés pour réduire localement l'erreur.

Contrairement aux techniques précédentes qui mettent en évidence des artefacts géométrique, c'est-à-dire des erreurs dans la préservation des proximités d'origine, Lespinats et Aupetit [146] introduisent une technique qui permet de révéler des artefacts topologiques, c'est-à-dire des erreurs de préservation du voisinage d'origine. Les artefacts topologiques se déclinent en deux types : des faux voisinages qui sont des points non voisins à l'origine dans les données et qui le deviennent sur la projection et des déchirures qui correspondent à des points voisins à l'origine mais non voisins sur la projection. Ces artefacts sont décrits plus en détails dans la suite. Ils proposent de visualiser deux mesures de stress simultanément, l'une indiquant les faux voisinages et l'autres les déchirures, à l'aide d'une échelle de couleur 2D perceptuellement uniforme. Un rayon de voisinage permet de définir le voisinage dans l'espace des données et dans l'espace la projection et peut être modifié manuellement ou automatiquement par une heuristique. Cette technique permet d'utiliser deux règles pour inférer des informations sur les données, à partir des contrastes de couleurs, pour des cas spécifiques (chevauchement des classes ou séparation de clusters), afin d'aider à repositionner virtuellement des points mal placés. Ils proposent également de colorer les cellules de Voronoï. Cette technique permet d'utiliser la projection malgré des erreurs de positionnement des points, mais les règles d'inférence sont limitées et la mesure de qualité ne garantie pas que les utilisateurs pourront exploiter la projection, en particulier pour une tâche de clustering visuel.

Des techniques interactives ont également été mises au point pour mettre en évidence ces artefacts topologiques. Très récemment, Martins et al. [155] a proposé différentes vues chacune permettant de révéler soit le stress global, soit des faux voisinages ou des déchirures selon différents niveaux de granularité (du point individuel aux groupes de points). Différentes mesures d'erreur spécifiques à chaque type d'artefacts sont proposées pour chaque vue et représentées par le biais d'une coloration interpolée entre les points ou d'arcs compactés (edge bundling) reliant les points voisins sur la projection pour montrer les voisins ou les points voisins à l'origine pour révéler les déchirures. Ces arcs sont également colorés selon l'intensité de l'erreur. L'objectif de cette approche est de permettre d'aider à mieux comprendre les effets des paramètres des algorithmes de projection en révélant différents niveaux de détails sur les artefacts géométriques et topologiques d'une projection. Afin d'éviter les effets d'occlusions dues aux arcs et obtenir des informations plus locales, l'utilisateur peut interactivement sélectionner un point ou un groupe de points (construit manuellement ou par clustering automatique) afin de ne visualiser que les artefacts qui leurs sont relatifs. De multiples vues statiques permettent de guider l'analyse visuelle des erreurs de projection et de les filtrer par la sélection d'un point ou d'un groupe.

Cette technique s'inspire de la *visualisation des proximités* [14] qui affiche interactivement en chaque point, en utilisant un échelle de niveaux de gris, les proximités d'origine dans l'espace des données relativement à un individu de référence sélectionné par l'utilisateur sur la projection. Cette technique affiche donc directement sur la projection le vecteur des proximités normalisées dans l'espace des données, que nous appelons proximités, c'est-à-dire la ligne de la matrice de similarité correspondant à l'individu référence. Les distorsions sont indirectement visualisées par le contraste entre les positions 2D et les couleurs représentant les proximités d'origine. La *visualisation des proximités* utilise une coloration des cellules de Voronoï en chaque point. L'intérêt de cette approche contrairement aux précédentes est de ne pas afficher une mesure de qualité, qui peut être biaisées par des points atypiques, mais afficher directement l'information d'origine, c'est-à-dire les proximités servant à projeter les données. Cette technique servira de base aux approches étudiées et développées dans la suite.

Qualité visuelle

D'autres mesures aident à définir la qualité d'une projection en utilisant des critères visuels particuliers comme les points isolés, les amas, les formes [257]. La qualité visuelle dépend de la tâche considérée. La plupart des mesures se concentrent sur la tâche de clustering visuel et quantifient la séparation visuelle de clusters prédéfinis. Ces mesures ne se limitent pas au projection, elles peuvent être étendues à d'autres visualisations telles que RadViz ou TableLens [5]. Les algorithmes de projection, comme les algorithmes de clustering automatiques, sont des boîtes noires utilisées de manière itérative : le choix et le paramétrage de l'algorithme sont modifiés itérativement jusqu'à obtenir une meilleur "qualité" par rapport à une métrique définie ou bien par rapport aux attentes de l'utilisateur et du modèle a priori qu'il a des données. Considérant une projection et un clustering des données en différentes classes, il y a plusieurs façons de définir et d'évaluer la qualité de la projection, dont voici les principales mesures :

Class Consistency Measure (CCM) [211] mesure la préservation de la proximités des individus au centroid de la classe auxquels ils appartiennent. Cette distance Euclidienne calculée sur la projection doit être plus faible que la distance aux autres centroids, mais cette contrainte n'est pas toujours respectée. Aussi cette mesure calcule le ratio du nombre de points qui violent cette contrainte pour indiquer la qualité de la projection.

Histogram Density Measure (HDM) [221] mesure l'entropie, c'est-à-dire l'information moyenne, de différentes portions de la projection découpée en grille. La projection est découpée en carrés (bin) et on compte dans chaque carré le nombre de points ainsi que l'étiquette de leur classe. L'entropie de chaque carré se calcule comme suit : $H(p) = -\sum_c \frac{p_c}{\sum_c p_c} log_2 \frac{p_c}{\sum_c p_c}$, avec p_c le nombre points de la classe c et lorsque tous les points sont de la même classe $H(p) = 0$.

Class Density Measure (CDM)[221] mesure le chevauchement entre classes. Pour se faire on utilise une représentation de chaque classe, c'est-à-dire pour une classe donné, on ne conserve que les points de la projection appartenant à cette classe et on crée ainsi une image. On calcule ensuite une fonction de densité continue basée sur les voisinages locaux. Sur chaque image, on calcule en chaque pixel la distance Euclidienne à son k-ème plus proche voisin de la même classe et la densité locale est obtenue en prenant l'inverse de cette distance. Pour obtenir la mesure de qualité, on estime ensuite le chevauchement mutuel en calculant la somme de la valeur absolue des différences entre pixels.

Toutefois une évaluation de ces trois mesures [222] a révélé qu'elles ne permettaient pas à elles seules de faire la différence entre les "bonnes" et les "mauvaises" projections. La taxonomie des facteurs de séparabilité des clusters de Sedlmair [198] a également mis en évidence que ces mesures automatiques ne rivalisaient pas encore avec les capacités humaines de jugement et qu'il était difficile de résumer la complexité visuelle d'une projection. Ces mesures se basent sur un clustering prédéfini des données afin de juger si la projection reflète correctement ce clustering. Cependant la notion de cluster est en soit difficile à définir [121], car elle dépend du domaine d'application et des propriétés intrinsèques aux données.

D'autres approches existent pour essayer contrôler et de mettre en évidence un clustering des données avec une projection. Elles reposent sur la prise en compte de contraintes spécifiées par l'utilisateur soit dans le processus de projection, soit sur la définition de la matrice de similarité. Par exemple, Local Affine MDS (LAMP)[124] ou une ACP sous contraintes [154] proposent d'adapter

la projection à des contraintes de position sur des points ayant été interactivement positionnés par l'utilisateur afin de prendre en compte des connaissances expertes. Ces contraintes de position peuvent également être prises en compte dans la matrice de similarité [152] ou directement dans la pondération des dimensions utilisées par la mesure de similarité [39]. Déplacer manuellement des points de la projection permet de dynamiquement définir ces contraintes mais on peut également spécifier des contraintes par rapport au clustering [40, 38]. Mais ces approches ne permettent pas de s'abstraire des problématiques d'artefacts de projection car elles ne garantissent pas l'optimalité dans le contrôle de la position des points.

Le processus de projection est donc difficile à contrôler. La section suivante présente plus en détails chaque étape de ce pipeline de réduction de dimension, ainsi que l'impact des artefacts de projections sur les usages et tâches associées aux projections.

2.3 Pipeline de réduction de dimension

Visualiser et analyser des données de grande dimension nécessite d'abstraire les variables pour se concentrer sur les relations de similarité entre individus et en particulier sur les structures sous-jacentes aux données qui en résultent. On considère ainsi soit directement un tableau individus/variables, soit directement une matrice de similarité (ou de dissimilarité) entre individus obtenue pour une certaine mesure de similarité (ou de dissimilarité). Différentes opérations sont nécessaires pour passer des données brutes à la visualisation. La section suivante décrit ce processus de réduction de dimension en précisant les biais introduits à chaque étape.

2.3.1 Description du pipeline

Le modèle de référence de la visualisation d'information ne met pas en valeur les biais qui sont introduits à chaque étape du processus. Nous ne nous intéressons pas ici au problème de visualisation des incertitudes, problème pour lequel le pipeline de visualisation à déjà été adapté [166, 58], mais nous nous intéressons aux biais introduits à chaque étape du pipeline de visualisation. En particulier, nous nous plaçons dans un contexte où l'on souhaite visualiser des données par réduction de dimension et nous précisons quels biais sont contrôlables en ayant recours à l'interaction de l'utilisateur, quels biais sont difficilement maîtrisables et quels biais nous nous proposons d'aider à maîtriser, en l'occurrence les artefacts de projection (Figure 2.10).

Acquisition Le pipeline de visualisation commence avec une source de données. Les données représentent un phénomène observé à travers un certain système de mesure. Collecter les données dépend du domaine d'application, mais dans tous les cas la qualité de l'information sous-jacente aux données est dépendante de la précision du système de mesure et du calibrage de celui-ci. Ce processus d'*acquisition* permet d'obtenir des *données brutes* qui devront ensuite être pre-traitées pour être nettoyées de leur *bruit* : filtrer les aberrations, détecter les valeurs manquantes, etc., dans le but d'obtenir des données structurées sous une forme canonique [45]. Le calibrage du système de mesure par un expert permet de contrôler la qualité de l'acquisition et ainsi de maîtriser en partie le bruit introduit à cette étape du pipeline.

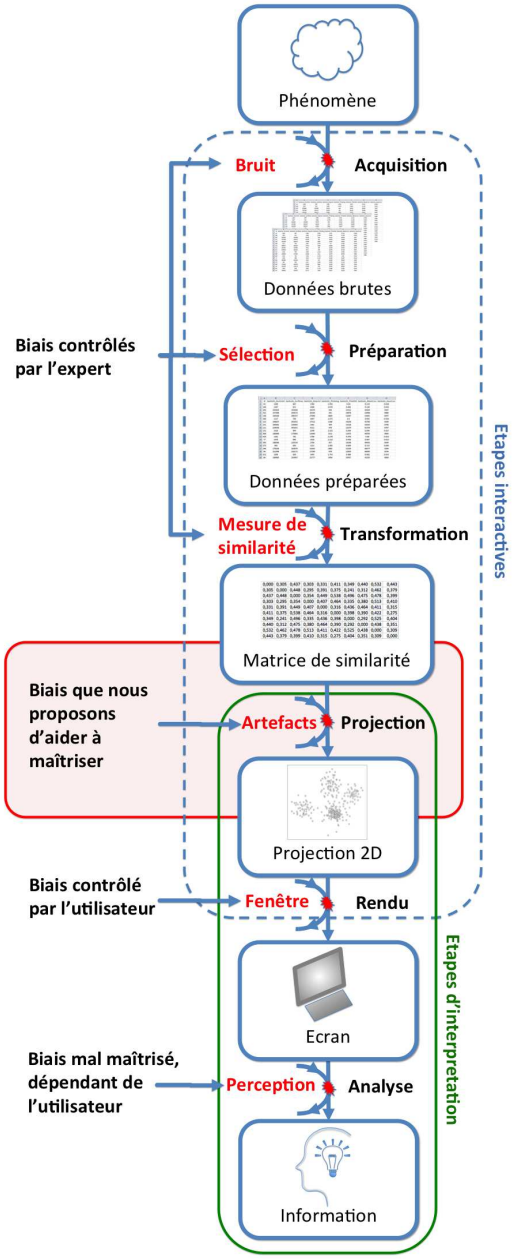


FIGURE 2.10 – Pipeline de réduction de dimension. Chaque étape du pipeline introduit un biais (en rouge). Les étapes interactives (entourées en pointillés bleus) sont des processus que l'utilisateur peut contrôler. On distingue des processus que seul un utilisateur expert peut paramétrer, des étapes d'interprétation (entourées en vert) liées à l'exploration et l'analyse de la projection par un utilisateur non nécessairement expert. Nous nous intéressons ici à l'étude du biais introduit par les artefacts de projection et proposons des outils afin d'aider à maîtriser ce biais afin de rendre plus fiable l'analyse visuelle d'une projection de données.

Dans certains domaines d'application, comme l'imagerie médicale, les experts s'opposent à l'application de prétraitements comme le filtrage ou l'équilibrage des données, de peur de perdre des informations importantes ou d'ajouter des artefacts. Il est souvent préférable de visualiser directement les données brutes sans prétraitements de manière à localiser les données manquantes, ou déceler des erreurs de calcul ou de paramétrage. La prise en compte des métadonnées comme les unités d'échelle ou de mesure sont également importantes pour paramétrer correctement le prétraitement des données. Détecter et résoudre les problèmes de données manquantes est un traitement souvent nécessaire lorsque l'on travaille sur des données réelles. Ces anomalies peuvent provenir de capteurs défaillants, d'erreurs de saisie, ou de l'absence même d'observations mesurables.

Différentes stratégies sont possibles pour nettoyer les données : retirer un individu (ou une dimension) selon le nombre d'observations manquantes, assigner une valeur en dehors de l'intervalle de définition de la dimension (comme -1 par exemple sur une dimension positive) de manière à visualiser les erreurs, assigner une valeur moyenne pour ne pas impacter les statistiques mais cela peut masquer certain outliers, assigner la valeur d'un individu très similaire (mais l'individu le plus similaire sur un sous ensemble de dimensions n'est pas forcément son plus proche voisin lorsqu'on considère l'ensemble des dimensions), ou enfin assigner une valeur par l'intermédiaire d'un modèle statistique. Après cette étape de prétraitement, les données brutes sont au format de table, où les lignes correspondent à des individus et chaque colonne correspond à une dimension caractérisant chaque individu par une valeur numérique.

Préparation En fonction des besoins associés au domaine d'application et à l'étude du phénomène observé, les lignes et colonnes, de la table des données, peuvent être filtrées pour retirer les individus atypiques et les variables non pertinentes. Ce filtrage peut être fait de manière automatique par des algorithmes de sélection de variables [157] ou manuellement selon une mesure de qualité [123] en interagissant avec la visualisation pour contrôler l'ordonnancement des dimensions par qualité et les filtrer manuellement afin de réduire leur quantité. A partir d'une projection de données, on peut également effectuer manuellement la sélection et le filtrage d'individus atypiques, aussi appelés outliers de données [2]. Le processus de transformation des données brutes permet d'obtenir une table de *données préparées* qui seront ensuite associées à une forme visuelle [45]. Cette étape de préparation est biaisée par le choix de l'approche et du critère de *filtrage/sélection*.

La normalisation est également un prétraitement fréquent notamment lorsqu'on calcule une mesure de similarité sur les données. Ce procédé permet d'obtenir des dimensions ayant le même intervalle de définition ($[0,1]$ par exemple). Ceci permet de rendre les dimensions comparables dans le calcul d'une distance Euclidienne entre individus. On peut ensuite avoir recours à d'autres transformations, comme l'application du logarithme, afin d'équilibrer les distributions et éviter des effets de dominance de certaines dimensions dans le calcul de la distance. Chaque dimension est normalisée séparément selon des bornes à définir, comme les minimums et maximums sur l'ensemble des individus, ou bien selon une borne relative à l'unité de mesure. Si nous considérons des données numériques $P \subseteq \mathbb{R}^n$ où chaque individu $p_i \in P$ est caractérisé par un ensemble de dimensions : $p = [a_1, a_2, \ldots, a_n]^T$, la normalisation de la dimension a_1 de l'individu p s'écrit : $p[a_1] = \frac{p[a_1] - min(a_1)}{max(a_1) - min(a_1)}$.

Transformation Cette étape nécessite l'introduction d'une *mesure de similarité* entre les individus afin de définir un *espace des données*. Cette espace se matérialise par une *matrice de similarité*. La taille de la matrice de similarité dépend du nombre d'individus considérés. La *mesure de similarité* définit les *relations de proximité* entre individus et cette transformation constitue un biais dans le sens où elle donne un certain point de vue sur les données. En effet, comme nous l'avons évoqué précédemment, le choix de cette mesure dépend du domaine d'application et la sémantique sous-jacente à cette mesure impacte directement l'interprétation des structures qu'elle induit dans les données, c'est-à-dire la caractérisation et l'interprétation des clusters dans les données.

Cette sémantique dépend directement du choix des dimensions pris en compte par cette mesure. On appelle *sous-espace* (respectivement *échantillon*) l'ensemble des dimensions (respectivement individus) sur lesquelles la mesure de similarité est calculée. Par exemple, on peut chercher un sous-espace qui optimise la séparation entre clusters [137]. Le sous espace peut être crée automatiquement [223] ou manuellement directement à partir de la projection [39]. Une fois la matrice de distances calculée, les données peuvent être projetées selon une approche linéaire ou non linéaire. Il est à noter que pour l'Analyse en Composantes Principales (ACP), on utilise la matrice des covariances mais la sémantique sous-jacente à cet espace des données est la même que celle d'une projection "Classical MDS" avec une distance Euclidienne sur l'ensemble de dimensions.

Projection L'étape de projection est centrale car elle permet d'obtenir une représentation des données selon la métaphore "proche $\approx$ similaire". Même si les algorithmes de projection tendent à préserver au mieux les structures sous-jacentes existantes dans l'espace des données, toutes les distances Euclidiennes dans l'espace de projection 2D ne respectent pas parfaitement les proximités d'origine. Comme nous l'avons vu précédemment, différentes mesures existent pour quantifier la qualité de la projection, c'est-à-dire la proportion de points et groupes de points mal placés. Ces artefacts de projection distordent les structures d'origine et impactent ainsi la fiabilité des inférences faites à partir de la projection.

On distingue deux catégories d'artefacts : les artefacts géométriques [14] et les artefacts topologiques [146]. Les artefacts géométriques sont causés par de faibles distorsions des distances d'origine comme des compressions et des étirements. Les distances 2D sont alors imprécises mais les voisinages sont préservés : ces artefacts n'impactent pas directement la fiabilité de la projection. Lorsque les distorsions des proximités sont trop importantes, la topologie d'origine n'est plus respectée et on parle alors d'artefacts topologiques : les voisinages 2D ne correspondent plus à ceux d'origine, des points proches sur la projection correspondent à des individus dissimilaires (faux voisinages) et des points éloignés correspondent en réalité à des individus similaires dans l'espace des données (déchirures). Ces artefacts sont très problématiques car ils déforment la structure sous-jacente aux données et amènent à de fausses interprétations lors de l'analyse de la projection comme des erreurs sur le clustering des données.

On considère un rayon r^* qui définit le voisinage autour d'un individu de référence p_s^* dans l'espace des données et un rayon r qui définit le voisinage radial du point de référence correspondant p_s dans l'espace de projection. On distingue alors 3 types de points relatifs à l'individu de référence (Figure 2.11) :

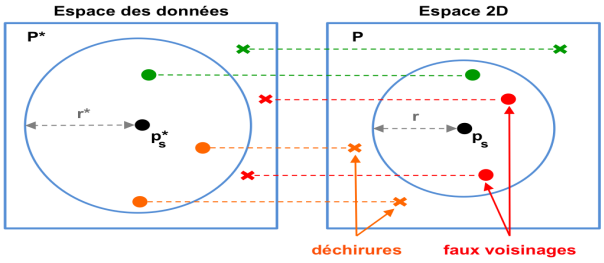
Voisins : Points qui sont voisins du point de référence à la fois dans l'espace 2D et dans l'espace des données.

Artefacts : Points qui sont voisins du point de référence dans un espace mais pas dans l'autre. Deux différents artefacts topologiques peuvent être distingués [146] :

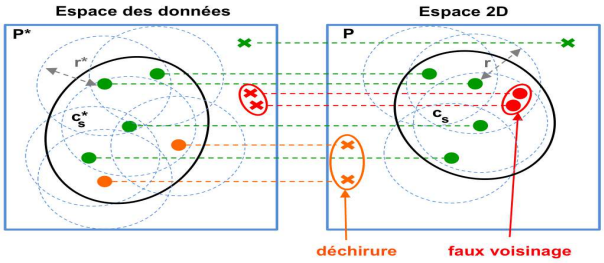
- *Faux voisinages :* Des individus dissimilaires sont associés à des points proches sur la projection 2D : $P_f = \{p_f \in P \mid d^*_{s,f} \geq r^* \text{and } d_{s,f} \leq r\}$

- *Déchirures :* Des individus similaires sont associés à des points éloignés sur la projection 2D : $P_t = \{p_t \in P \mid d^*_{s,t} \leq r^* \text{and } d_{s,t} \geq r\}$

Autres : Les points non voisins de la référence dans un espace et dans l'autre.



(a) Artefacts topologiques avec la granularité d'un individu



(b) Artefacts topologiques avec la granularité d'un cluster

FIGURE 2.11 – Schéma illustrant les différents types d'artefacts topologiques : déchirures (orange) et faux voisinages (rouge). Cette définition par rapport à un individu (a) se transfère par rapport à un cluster d'individus (b). Dans chaque espace, les points représentés par des cercles correspondent à des voisins de la référence, contrairement aux points représentés par des croix. Les points "biens placés" sont représentés en vert.

Les artefacts topologiques ont deux caractéristiques importantes : relativité et granularité. En effet, ils sont relatifs à une référence. Par exemple, un point qui est faux voisin pour ses voisins 2D peut être une déchirure pour ses voisins d'origine. Ensuite, on peut définir ses artefacts à différents niveaux de granularité : les erreurs de voisinage relatives à un individu de référence se transfèrent à un cluster d'individus. Par exemple, deux clusters distincts dans l'espace des données peuvent être projetés au même endroit sur la projection et ainsi définir une région de faux voisinages. De la même façon, un cluster peut être déchiré en différentes composantes non connexes sur la projection et chaque composante prise séparément est alors une déchirure par rapport aux autres. Morceler les clusters impacte "seulement" la fiabilité des inférences faites à partir de la projection, mais les faux voisinages impactent également la séparation des clusters sur la projection ce qui nuit au confort de son analyse visuelle. Les artefacts topologiques impactent les tâches d'analyse visuelle à différents niveaux, comme nous le verrons plus en détail dans la suite.

Rendu Dans la dernière étape, l'espace de projection est représenté sous la forme d'un nuage de points 2D, où chaque point représente un individu et où les axes n'ont pas de signification (excepté dans le cas de l'ACP). Ce nuage de points est visualisé à l'écran au travers d'une fenêtre. L'espace à l'écran alloué pour cette visualisation impacte directement le confort de son analyse visuelle. En effet, les distances 2D entre les points sont directement liées à la largeur et hauteur de l'espace dédié à la vue de la projection.

Des transformations géométriques de la vue, contrôlées par l'utilisateur (pan and zoom), permettent de s'affranchir en partie de ce biais induit par une vue de taille fixe. Ces transformations peuvent être très utiles notamment pour le passage à l'échelle de la projection afin de pouvoir zoomer sur des zones avec de fortes occlusions. Un zoom géométrique présente la contrainte de perdre le contexte global de la projection. Les lentilles déformantes (Fisheye) [92] résolvent ce problème en affichant un niveau de zoom plus grand dans une région d'intérêt (souvent circulaire). Une zone de déformation sur les bords permet de faire la jonction avec le niveau de zoom du reste de la vue. Ces interactions avec la vue ne nécessitent pas une expertise forte de l'utilisateur.

Analyse L'utilisateur analyse le nuage de points rendu pour effectuer des tâches d'analyse visuelle et extraire ou exploiter la structure sous-jacente aux données. La perception de la projection est biaisée par les capacités cognitives de l'homme (reconnaissance de formes, vision des couleurs, limites mémoire) [134, 100] et par l'encodage graphique choisi (points, taille, couleur). L'information décodée de cette visualisation est donc sujette à des erreurs de jugement.

Les analystes de données disposent de systèmes permettant de surveiller et contrôler ce pipeline de réduction de dimension, dans le but d'extraire au mieux l'information cachée dans les données. La section suivante présente différents systèmes d'analyse implémentant ce pipeline, puis les principaux usages de la visualisation par projection et enfin une taxonomie des tâches d'analyse visuelle des projections. Cette taxonomie est présentée en détaillant les possibles impacts des artefacts sur chaque tâche.

2.3.2 Taxonomie des usages du pipeline

En fonction de la tâche d'analyse à réaliser, le pipeline de projection peut être transformé ou rejoué en utilisant différentes configurations, projections, mesures de similarité, filtrages des données. Cependant les algorithmes de projection sont des boîtes noires dont la sensibilité est étroitement liée aux propriétés intrinsèques des données. Les analystes de données ne peuvent donc pas contrôler parfaitement le résultat du pipeline. Ils doivent être informés des biais de ce pipeline et en particulier des problématiques d'analyse visuelle liées aux artefacts de projection.

Systèmes d'analyse basés sur les projections

Nous présentons ici les principaux systèmes qui permettent de contrôler le processus complet de réduction de dimension, ainsi que les techniques interactives existantes pour l'aide à l'analyse visuelle des projections.

Projection Pursuit Guided Tour [56] permet de visualiser interactivement une série de projections pertinentes de manière animée afin de suivre le contexte d'une projection à l'autre. L'utilisateur peut interagir afin de visualiser d'observer la transformation de la projection par rapport aux axes. Cette technique est une partie d'un système d'analyse exploratoire, avec des vues coordonnées, permettant de contrôler le processus d'optimisation de la projection et suivre l'évolution du stress ainsi que d'assister un clustering manuel ou automatique des données afin de comparer différentes approches. Ce système a ensuite été amélioré au cours du temps avec différentes versions : XGobi [216], XGvis [41], puis Ggobi [57, 217].

Toutefois ce système ne guide pas l'utilisateur dans la configuration du processus complet, contrairement à DimStiller [117] qui a été conçu pour des utilisateurs novices en analyse exploratoire de données de grande dimension. Ce système propose à l'utilisateur de configurer et chaîner interactivement différents opérateurs permettant de passer d'une table des données à un nuage de points en proposant des retours sur la qualité de chaque étape du processus de réduction de dimension afin de guider l'utilisateur dans son paramétrage. Les différentes visualisations sont liées entre elles, mais les interactions se limitent à la sélection par brossage et aucune aide n'est apportée pour interpréter les projections proposées.

La technique Spider Cursor [225] affiche interactivement le graphe des k-plus proches voisins directement sur la projection. Les voisins dans l'espace des données d'un point sélectionné sur la projection sont chacun reliés à ce point par un arc. Cette technique est similaire à la visualisation interactive des proximités [14] mais l'affichage des arcs pose des problèmes de croisements et d'occlusions d'autant que le graphe des k-plus proches voisins en grande dimension est très dense. Cette technique fait partie du système ProjEx [168] proposant une grande variété d'algorithmes de projection et des outils pour aider à paramétrer ces algorithmes.

D'autres techniques existent également pour visualiser statiquement les proximités entre neurones dans des cartes auto-organisées (SOM) comme la U-Matrix [236] qui colore chaque neurone selon sa distance moyenne aux autres neurones, CONNvis [220] qui connecte par des liens les neurones voisins, ou les Component Plane [246] qui affichent la distribution d'une dimension directement sur le SOM par le biais d'une échelle de couleur monochromatique, de manière à mettre en valeur un possible clustering des données sur une dimension.

Excepté DimStiller, ces différents systèmes s'adressent principalement à des experts en analyse de données. Ils incorporent des techniques permettant d'aider et guider le travail d'analyse visuelle de ces experts. Mais assez peu d'attention est portée aux problématiques liées aux artefacts de projection. Or ces artefacts peuvent influencer la fiabilité des interprétations faites à partir des projections. Ces interprétations dépendent d'une part du contexte d'utilisation de la projection et d'autre part de la tâche d'analyse visuelle considérée.

Usages de la visualisation par projection

Les projections de données sont aujourd'hui principalement utilisées par des experts en analyse de données. Le recours à cette visualisation intervient à différentes étapes d'un processus plus large de traitement de données de grande dimension. Les analystes de données l'utilisent pour obtenir un premier aperçu de la qualité des données avant le paramétrage d'un algorithme de classification supervisée ou non supervisée. Cet aperçu leur permet de se faire une idée de la complexité du problème de clustering, ou de classification, par rapport aux données considérées.

Le problème de classification revient à classer les individus dans différents groupes d'individus similaires (clusters), c'est-à-dire à attribuer à chaque individu une étiquette correspondant au cluster auquel il appartient. On note *cluster* un groupe de données similaires défini sans étiquettes a priori et *classe* un groupe de données partageant la même étiquette. Selon la prise en compte ou non d'un modèle a priori des données, c'est-à-dire la prise en compte d'étiquettes pré-définies sur tout ou une partie des données, on distingue différents contextes de classification [25] :

Classification supervisée : on cherche à attribuer une étiquette de classe à de nouvelles données à partir de données déjà étiquetées. Pour cela on entraine un classifieur qui va apprendre à "reconnaître" les classes des données par rapport à leurs caractéristiques et permettre d'attribuer l'étiquette de classe la plus pertinente à une nouvelle donnée.

Classification non-supervisé : on cherche à construire une partition des données en différents clusters sans information d'étiquettes a priori. Le paramétrage et les modèles utilisés par ces algorithmes affectent la taille de la partition des données.

Classification semi-supervisé : on exploite les données déjà étiquetées et les données non étiquetées afin d'entrainer un classifieur.

L'utilisation de la projection en amont d'une classification supervisée permet de vérifier si les données peuvent être classifiées linéairement. En effet, si l'Analyse en Composantes Principale dévoile sur la projection des clusters 2D nettement séparés correspondant à chacune des classes, alors le problème de classification peut être résolu facilement par des approches linéaires. L'étude de la connectivité entre les classes [67], c'est-à-dire des relations de proximité et de frontières entre classes, permet également aux analystes de cibler quelles classes seront plus difficiles à séparer entre elles. En fonction des proximités entre classes visibles sur la projection, les analystes peuvent ainsi décider de diviser le problème de classification automatique en plusieurs étapes. Mais la présence d'artefacts peut induire en erreur l'analyste dans ses inférences sur les proximités dans l'espace des données.

La projection est également intéressante en amont d'une classification non-supervisée. L'analyse du clustering d'un jeu de données n'est pas une tâche entièrement automatique, mais plutôt un processus itératif visant à extraire des connaissances et estimer leurs qualité par rapport à des attentes précises. Le clustering automatique permet de synthétiser les données afin soit de réduire le nombre de données à analyser, soit d'aider à modéliser le comportement du système ou du phénomène à partir duquel les données ont été acquises. Cette synthétisation peut amener à perdre des détails, à la manière de la compression de données, mais son but principal est de mettre en exergue les principales structures sous-jacentes aux données, à savoir principalement les différents clusters existants ainsi que leurs proximités et frontières définissant la topologie sous-jacente aux données.

Il existe une littérature très riche d'algorithmes de clustering automatique [122, 22, 121] utilisant différentes approches et critères pour séparer ou agréger les individus en clusters. Mais ces algorithmes sont des boîtes noires [219] et le recours à la projection peut permettre d'obtenir un aperçu du résultat de ces algorithmes afin d'aider à les paramétrer ou pour avoir confiance dans la pertinence du résultat. Mais de part la présence d'artefacts sur la projection, elle ne constitue pas un support fiable pour arriver à ces fins.

La détection de données atypiques (outliers) est également une tâche importante en analyse de données. Ces outliers peuvent correspondre à des anomalies introduites par le mécanisme d'acquisition, à une déviation dans le comportement du système observé, à des changements dans ce système, à des opérations frauduleuses, ou encore à une erreur humaine. Leur détection est primordiale pour identifier rapidement des erreurs ou altérations dans le système avant que leurs conséquences ne soient trop importantes. Il existe un grand nombre d'algorithmes automatiques de détection d'outliers [105]. Là encore le recours à la projection peut permettre de visualiser le résultat de cette détection automatique afin que l'analyste soit plus confiant. Les algorithmes de projection sont sensibles aux outliers et ils les mettent relativement bien en évidence, mais les problématiques d'artefacts ne garantissent pas que la projection révèle correctement tous les outliers sur la projection, ni qu'elle ne révèle pas de faux outliers.

L'utilisation de la projection comme support pour contrôler l'état d'un système ne s'adresse pas uniquement à des analystes de données. La métaphore de proximité sur laquelle reposent les projections étant relativement intuitive, le recours à la projection peut également se justifier pour des utilisateurs n'étant pas nécessairement intéressées par les enjeux d'analyse de données. Par exemple, ces non-experts peuvent utiliser la projection pour organiser leur exploration des données et naviguer dans une collection d'images [266]. Le contrôle de l'état d'un système peut également être effectué par des non-experts, dont la tâche est alors de valider que la structure sous-jacente aux données correspond bien au modèle indiqué sur la projection par des étiquettes de classes. Une déviation du système est supposée modifier la structure sous-jacente aux données et l'utilisateur remarque alors sur la projection la présence de nouveaux clusters ou bien un chevauchement entre classes qui n'existait pas précédemment dans d'autres données. L'intérêt de la projection est de permettre de visualiser directement les données brutes associées à ses éléments atypiques afin d'interpréter rapidement la situation.

Les proximités entre classes sont également une information qui peut renseigner des non-experts dans un contexte d'aide à la décision. Par exemple, les douaniers à la frontière contrôlent le passage des camions [7]. Un scanner du camion permet de retourner des informations sur la cargaison, sous la forme de signaux multidimensionnels, sans avoir à ouvrir le conteneur. A partir

de ces données de grande dimension, les douaniers doivent décider si le camion peut passer ou si il doit être ouvert et fouillé en détails. Cette décision est critique et dépend du jugement du douanier. L'utilisation d'une projection des données de scanner, avec d'autres données de cargaisons étiquetées, permet d'étudier localement les proximités des différentes mesures à celles déjà étiquetées comme correspondant à des éléments à risque, comme de la drogue ou des armes. Le douanier doit ensuite décider à partir de son analyse visuelle de la projection, si il juge nécessaire la fouille de la cargaison. Ce scenario amène à la question de recherche : les projections peuvent elles être utilisées par des non-experts pour de l'aide à la décision ? Ces utilisateurs non-experts ne cherchent pas à extraire des informations relatives à la classification, comme le font les analystes de données, mais plutôt à exploiter directement ces informations pour répondre à des problématiques de recherche d'information impliquant d'explorer les données, ou des problématiques de confirmation pour contrôler l'état d'un système et prendre des décisions. Néanmoins, si la projection ne respecte pas fidèlement la structure sous-jacente d'origine, ces applications peuvent être remises en question.

Les artefacts de projection sont donc problématiques dans l'analyse visuelle des projections, car ils remettent en cause la fiabilité des inférences faites sur les données à partir de la projection. La section suivante présente les différentes tâches d'analyse visuelle que peuvent effectuer des analystes de données ou des non-experts sur des projections. Nous expliquons également pour chaque tâche les possibles biais introduits par les artefacts.

2.4 Taxonomie des tâches d'analyse visuelle des projections

A ce jour, il n'existe qu'une seule taxonomie des tâches associées à la visualisation par réduction de dimension [196]. Cette taxonomie cible les usages des analystes de données et distingue, pour une mesure de similarité donnée, deux objectifs : l'analyse des structures sous-jacentes aux données et l'analyse des dimensions. Dans cette thèse, nous considérons une mesure de similarité fixée et nous nous consacrons uniquement à l'analyse des structures sous-jacentes introduites par cette mesure.

Indépendamment de l'expertise des utilisateurs en analyse de données, on distingue deux contextes d'application : exploratoire et confirmatoire. Les tâches du premier contexte consistent à détecter des *groupes implicites*, c'est-à-dire extraire la structure sous-jacente aux données, alors que les tâches du second contexte consistent à valider des *groupes explicites*, c'est-à-dire valider des étiquettes de classes. On distingue les tâches locales, qui concernent l'analyse des proximités relatives à une référence, des tâches globales qui reposent sur l'analyse des proximités au sein des structures sous-jacentes aux données, ainsi qu'entre ces structures dans l'espace des données.

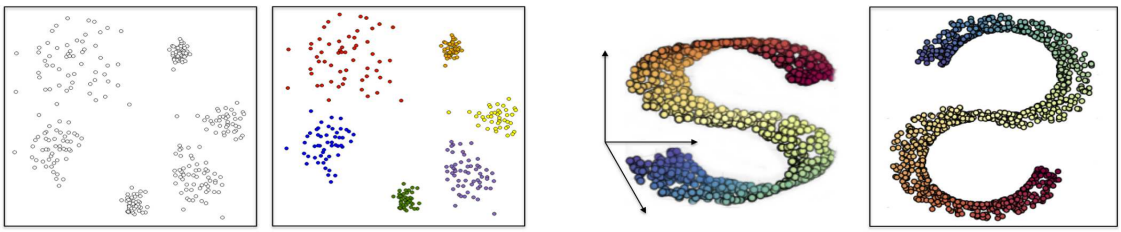
Nous introduisons dans cette section une nouvelle taxonomie des tâches d'analyse visuelle des projections qui prend en compte l'impact des artefacts de projection sur la fiabilité des inférences faites sur les données à partir de la projection et relativement à chaque tâche.

2.4.1 Analyse Exploratoire

Dans ce contexte, les données ne sont pas étiquetées, c'est-à-dire qu'aucune supposition n'est faite sur le modèle à l'origine des données. Le nuage de points 2D est alors visualisé sans couleurs sur les points (Figure 2.12-a). L'objectif est d'extraire des informations sur le clustering à partir de la projection [232]. Les proximités entre points et les motifs apparaissant en 2D sont les seules informations accessibles sur l'espace des données. Les tâches d'analyse visuelle de la projection permettent théoriquement d'inférer l'existence d'outliers et de clusters, ainsi que leurs relations de proximités, dans l'espace des données. Mais la présence d'artefacts de projection remet en cause la fiabilité de ces inférences. On distingue trois tâches d'analyse visuelle de la projection dans un contexte exploratoire :

Détection d'outlier : Détecter des points 2D isolés sur la projection pour inférer la présence d'un individu atypique, c'est-à-dire un individu n'appartenant à aucun cluster [212]. Les algorithmes de projection sont sensibles aux outliers de données et tendent à les mettre en évidence. Cependant des distorsions géométriques peuvent rendre les points moins visibles et rien n'exclut la possibilité que la projection masque certains outliers en les positionnant comme des faux voisinages au sein d'un cluster 2D (Figure 2.13-A1) ou inversement comme des outliers 2D alors qu'ils correspondent à des déchirures (Figure 2.13-A2).

Clustering visuel : Identifier les groupes de points proches entre eux sur la projection, c'est-à-dire des motifs de densité distincts, comme des "blobs" ou des formes géométriques (Figure 2.12-b), dans le but de définir un clustering de la projection devant correspondre théoriquement au clustering dans l'espace des données. Si le phénomène observé présente différents états connus a prori, alors la séparabilité des clusters sur la projection donne implicitement des retours sur la qualité de la mesure de similarité, c'est-à-dire sur sa capacité à capturer les propriétés du phénomène observé. Typiquement, l'absence de clusters remet en question le choix de la mesure de similarité. A l'inverse dans le cas par exemple d'une ACP, des clusters bien séparés indiquent que le problème de classification est linéaire, ce qui permet ensuite d'utiliser des outils linéaires de classification automatique. Cependant la présence de faux voisinages (respectivement déchirures) peut réduire (respectivement augmenter) artificiellement le nombre de clusters visibles en 2D par rapport à la vérité dans l'espace des données (Figure 2.13-B1-B2).



(a) Exploratoire (à gauche) et confirmatoire (à droite) (b) Cluster avec une forme géométrique en 2D

FIGURE 2.12 – Exemple d'une projection dans un contexte exploratoire et confirmatoire (a). Exemple d'un cluster représenté par une forme géométrique en 2D contrairement à des "blobs", ce qui donne une indication sur la complexité de sa topologie sous-jacente (b).

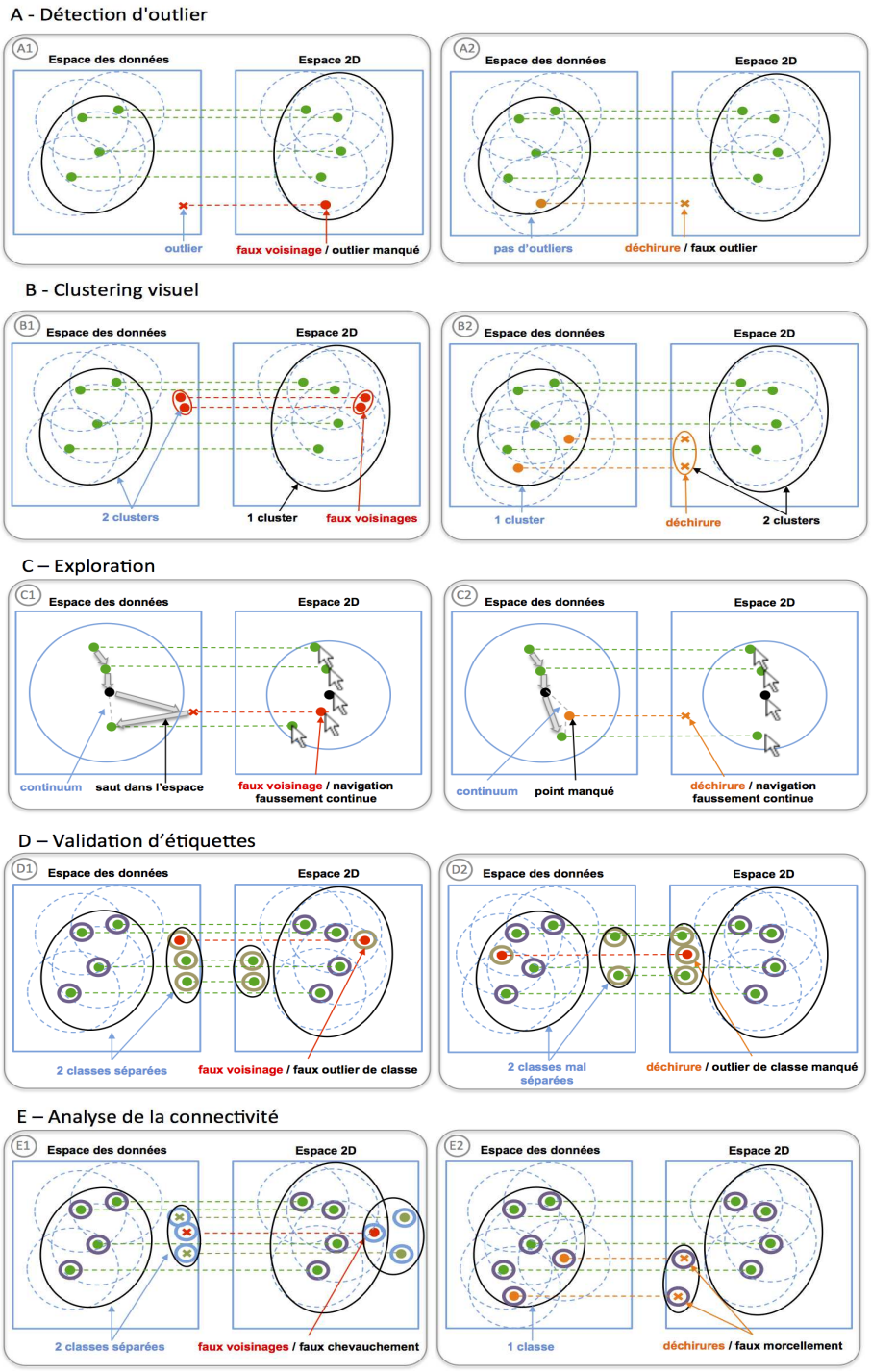


FIGURE 2.13 – Schémas illustrant les problèmes liés aux différents types d'artefacts topologiques (déchirures en orange et faux voisinages en rouge) sur les tâches d'analyse visuelle des projections. Les points "biens placés" sont représentés en vert. Les légendes en bleu indiquent la vérité dans les données, alors que les légendes en noir indiquent le biais d'inférence introduit par les artefacts. Dans les cas où on ne considère qu'un seul cluster dans les données, dans chaque espace, on représente les points dans le cluster par des cercles et ceux en dehors par des croix.

Exploration : Utiliser la projection comme support pour guider la navigation dans les données. Contrairement aux deux tâches précédentes qui ciblent des analystes de données, cette tâche concerne plutôt des non-experts. La projection permet une exploration des données d'individus similaires en individus similaires, ce qui fournit un ordre, ou tout du moins un guidage, pour visualiser les données brutes une par une, en garantissant peu de changements de l'une à l'autre le long de la trace d'exploration. Ce type d'exploration est pertinent lorsqu'on doit analyser manuellement chaque image d'une collection dans un cadre de recherche d'information pour en extraire des caractéristiques ou valider la qualité des images. Cependant la présence d'artefacts de faux voisinages peut amener à naviguer sur des individus dissimilaires et ainsi introduire un changement imprévisible de contexte (Figure 2.13-C1). De plus, l'exploration complète d'un voisinage dans l'espace des données peut impliquer de pouvoir naviguer sur des déchirures (Figure 2.13-C2).

2.4.2 Analyse Confirmatoire

Dans ce contexte, les étiquettes sont connues pour chaque donnée et définissent un clustering de référence que l'on cherche à valider par comparaison avec le clustering 2D de la projection. Cependant la présence d'artefacts peut fausser cette comparaison. On notera que si le clustering dans l'espace des données ne reflète pas le modèle indiqué par les classes, alors on peut remettre en question soit le modèle, soit la mesure de similarité. Dans le cadre de cette thèse, on ne cherche pas à remettre en cause la mesure de similarité, aussi on considérera des erreurs d'étiquetage. Le nuage de points 2D est visualisé avec une couleur sur chaque point correspondant à son étiquette (Figure 2.12). Ce contexte s'applique aussi bien à des analystes de données qu'à des non-experts.

Validation d'étiquettes : Détecter les points 2D qui ont des étiquettes différentes de tous leurs voisins 2D. Ces points sont des erreurs potentielles d'étiquetage (outliers de classe) ou des points à la frontière de la classe, qui ont donc un fort potentiel d'incertitude sur l'étiquetage. Cependant à cause des faux voisinages on peut voir des outliers de classe qui n'en sont pas et réciproquement on peut manquer des erreurs d'étiquetage à cause de déchirures (Figure 2.13-D1-D2).

Analyse de la connectivité : Identifier les points d'une classe donnée et étudier les relations de proximité entre ces points et ceux d'autres classes. Par exemple, un fort chevauchement entre classes indique que la mesure de similarité ne sépare pas correctement les classes. A l'inverse une classe scindée en plusieurs composantes distinctes indique la présence de sous-clusters, c'est-à-dire un manque de cohésion au sein de la classe qui peut impliquer la nécessité d'un étiquetage en sous-classes ou bien signifier un problème de la mesure de similarité. Si les clusters 2D correspondent aux classes et sont bien séparés sur la projection, alors la mesure de similarité est bien choisie et les données sont fidèles à leur modèle. Cependant on ne peut pas savoir si les problèmes de chevauchement (respectivement morcellement) des classes ne sont pas dus à des faux voisinages (respectivement déchirures) (Figure 2.13-E1-E2).

Donc indépendamment du contexte de l'analyse visuelle, les artefacts de projection sont susceptibles d'impacter très fortement les inférences effectuées sur l'espace des données à partir de la projection et d'amener à des conclusions erronées. Nous étudions, dans les chapitres suivants, une approche interactive basée sur une coloration de la projection en fonction des proximités d'origine à un individu de référence [14], afin d'une part de mesurer l'impact des artefacts de projection sur les tâches d'analyse visuelle et d'autre part d'évaluer s'il est possible de s'en abstraire.

3

ProxiViz
la visualisation interactive des proximités revisitée

L'interprétation d'une projection est rendue problématique par la présence d'artefacts issus de la réduction de dimension. Visualiser ces artefacts lors de l'analyse visuelle de la projection peut permettre de rendre plus fiables les inférences faites sur l'espace des données. La visualisation interactive des proximités [14] est une coloration interactive de la projection en fonction des proximités d'origine relatives à un individu de référence. Cette technique met en évidence les artefacts de projection relatifs à un individu de référence. Dans ce chapitre, nous discutons les enjeux d'encodage visuel et d'interaction de navigation associés à cette technique dans le but d'améliorer son efficacité pour l'aide à l'analyse visuelle des projections.

3.1 Motivation

La visualisation interactive des proximités [14] est une coloration interactive de la projection en fonction des proximités d'origine relatives à un individu de référence. Cette technique dans sa version d'origine utilise une échelle de niveaux de gris et les cellules de Voronoï comme support de l'encodage visuel des proximités issues de l'espace des données. Ces proximités d'origine sont visualisées interactivement sur la projection par sélection à la souris du point de référence.

Le principe de cette technique est simple : utiliser la couleur comme variable visuelle en supplément de la position des points. Ces positions encodent des relations de proximités 2D approximant celles d'origine. La visualisation interactive des proximités encode directement les proximités d'origine, relatives à un individu de référence, sur la projection. Par comparaison des couleurs et positions, cette technique met ainsi en évidence les artefacts de projection relatifs à la référence.

Contrairement aux techniques existantes pour visualiser les artefacts, cette coloration permet également d'analyser visuellement les relations de proximités relatives à cet individu dans l'espace des données afin par exemple d'identifier ou valider un individu atypique. Cependant les proximités visualisées interactivement étant locales, l'inférence de clusters dans les données n'est pas directe et nécessite l'exploration de différentes références. De plus, la projection servant de support pour la coloration et l'interaction, cette technique n'est pas complétement indépendante des problématiques d'artefacts. En effet, la coloration est moins visible dans les régions denses de la projection et la sélection d'un faux voisinage, c'est-à-dire une référence non voisine de la précédente dans l'espace des données, peut impliquer un changement radicale des couleurs représentées (Figure 3.1). Nous utiliserons dans la suite le même jeu de données fil-rouge que dans l'introduction (Figure 1.4).

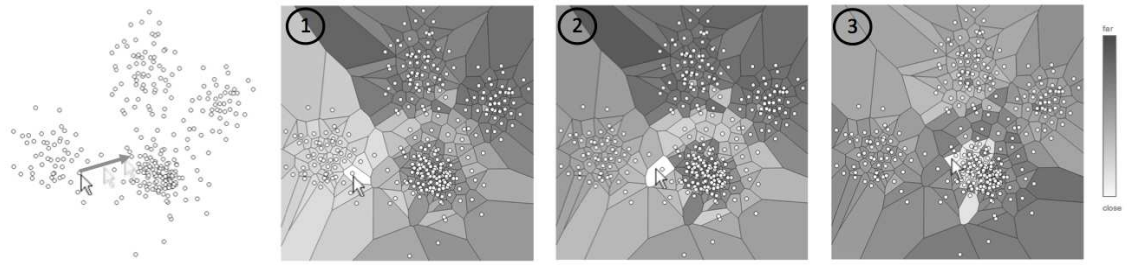


FIGURE 3.1 – Exemple de problème de sélection du point de référence dans le cas de faux voisinages. Nous reprenons l'exemple fil-rouge de l'introduction, à savoir la projection d'un jeu de données synthétiques composé de 6 Gaussiennes réparties aléatoirement sans chevauchements dans un espace de 10 dimensions. Plus la couleur est claire et plus l'individu associé à la cellule de Voronoï est proche de l'individu de référence dans l'espace des données. La référence correspond au point le plus proche du curseur de la souris sur la projection. On remarque que la coloration change radicalement pour les trois références sélectionnées consécutivement. Ceci s'explique par le fait que chaque référence appartient en réalité à un cluster différent dans l'espace des données. Cet exemple typique des faux voisinages.

Ces problématiques peuvent jouer un rôle significatif sur l'efficacité de la technique. Dans ce chapitre, nous présentons plus en détails les enjeux d'encodage visuel et d'interaction de navigation associés à la visualisation interactive des proximités. Nous proposons une nouvelle conception de cette technique, que nous appellerons dans la suite *ProxiViz*.

3.2 Encodage visuel

La matrice de similarité définit les relations de proximité entre individus dans l'espace des données. Pour un individu sélectionné, les proximités à cet individu de référence correspondent à une ligne de la matrice, c'est-à-dire à un vecteur de valeurs numériques indiquant pour chaque individu sa "ressemblance" à l'individu référence. Il existe différentes variables visuelles [120] permettant représenter ce vecteur et parmi les plus pertinentes pour des valeurs quantitatives, on trouve dans l'ordre décroissant d'efficacité : la position, la brillance de la couleur et la taille. Différents supports d'application sont possibles pour représenter l'encodage de ce vecteur, comme l'intérieur des points, leur contour, ou leur arrière-plan.

Dans la visualisation interactive des proximités [14], la brillance de la couleur est utilisée et appliquée aux cellules de Voronoï. La position des points sur la projection encode déjà les différentes approximations des proximités issues de la matrice de similarité. Aussi la brillance de la couleur est une bonne alternative pour ajouter interactivement une information plus locale et sans distorsions. L'échelle de couleur qui est utilisée dans la visualisation interactive des proximités est un dégradé de gris, où plus la couleur est sombre et plus la proximité est faible (grande distance), et où plus la couleur est claire et plus la proximité est grande (faible distance). Pour donner une idée de la distribution des proximités, l'échelle de couleur indique la proximité maximum et minimum dans la matrice de similarité, de même pour le vecteur des proximités à la référence (Figure 3.2). Avec l'indication supplémentaire de la moyenne, ces bornes permettent d'avoir une idée de la distribution des proximités visualisées en couleur par rapport à la distribution dans toute la matrice.

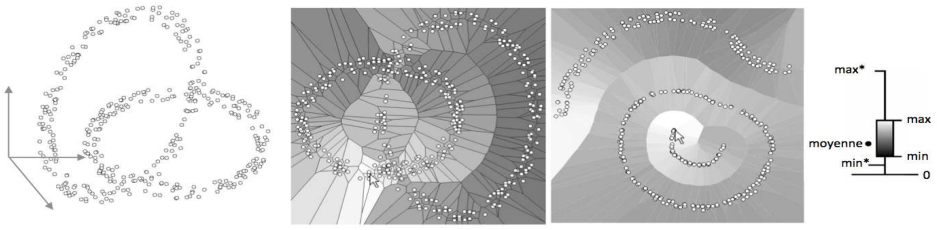


FIGURE 3.2 – Exemple de visualisation interactive des proximités [14] sur deux projections d'un jeu de données composé de deux anneaux entrelacés en 3D (à gauche). La projection ACP a introduit des faux voisinages (au centre) alors que la projection MDS a créé des déchirures (à droite). L'échelle de couleur en niveaux de gris présente des indicateurs de la distribution des proximités (min^*/max^* sur la matrice de similarité, min/max sur le vecteur des proximités à la référence et la moyenne).

3.2.1 Les échelles de couleur

La littérature sur l'étude des échelles de couleurs [209] est importante en visualisation scientifique, ainsi qu'en visualisation d'information et s'appuie sur des études en science cognitive et en colorimétrie. On distingue différents modèles de couleur selon le type de données à associer à une couleur. Définir un modèle de couleur revient à définir un système de coordonnées et un sous-espace dans ce système, où chaque couleur correspond à un point.

On distingue deux types de systèmes de coordonnées définissant un espace de couleur : les systèmes dépendants du dispositif d'affichage et ceux indépendants. Le système de couleur RGB se base sur la définition des couleurs par ajout des trois couleurs primaires : rouge, vert, bleu. Différentes variantes peuvent en être dérivées pour par exemple faire varier facilement la brillance, dont la variation est intuitivement perçue par l'Homme, avec le système HSV qui combine la teinte, la saturation et la brillance. Le système RGB est le plus utilisé, mais les couleurs peuvent varier sensiblement d'un dispositif à l'autre. C'est pourquoi des systèmes indépendants ont été développés tel que le système de couleur CIE LAB [85].

Une fois le système de coordonnées défini, il faut introduire un sous-espace de ce système de coordonnées de couleurs permettant de transformer une suite de valeurs, numériques ou non, ordonnées ou non, en une suite de couleurs, c'est-à-dire fonction de transfert des valeurs vers les couleurs. Ce sous espace est représenté par une échelle de couleur indiquant les différentes couleurs pour les valeurs possibles. Les échelles de couleur ont différentes propriétés [150] [231] :

l'ordre : les couleurs doivent induire un ordre perceptuel si les valeurs sont ordonnées et aucun ordre dans le cas contraire

l'uniformité : les écarts de distances perceptuelles entre les couleurs doivent représenter les mêmes écarts qu'entre les valeurs. Les couleurs doivent être clairement séparables dans le cas de valeurs nominales.

les frontières : des valeurs discrètes doivent correspondre à une échelle clairement séparée et des valeurs continues doivent être associées à une échelle qui donne une impression de continuité dans les couleurs

le principe de la diagonale : pour des données bi-variées, comme des températures, on doit pouvoir clairement faire la différence entre les couleurs qui représentent des valeurs au dessus et en dessous de zéro.

Il existe ensuite un certain nombre de règles pour la construction d'une échelle de couleur, afin de tirer au mieux profit de ses propriétés [177]. Par exemple, les échelles de couleurs, faisant varier la brillance pour une même teinte et une même saturation, sont perpétuellement efficaces pour représenter des valeurs quantitatives. Pour des valeurs nominales, l'échelle de couleur ne doit pas idéalement excéder sept couleurs pour rester perceptuellement efficace [151], car au delà on ne perçoit plus aussi bien les écarts entre couleurs et la mémorisation des valeurs associées aux couleurs devient difficile.

Le choix d'une échelle doit s'adapter à différents facteurs [140] [177] dont le type de données, la tâche devant être effectuée, ainsi que les utilisateurs et leurs habitudes. En effet, il existe des échelles de couleurs qui sont des standards dans certains domaines d'application, sans qu'elles aient forcément de justification au niveau de leur efficacité perceptuelle. Par exemple, l'échelle

de couleur arc-en-ciel est très utilisée en physique et pourtant il a été montré qu'elle n'était pas efficace car l'ordre des couleurs n'est pas intuitif [31]. Cependant elle reste une référence, car les physiciens ont appris à l'utiliser et ils peuvent continuer à comparer leurs résultats avec des résultats antérieurs. De plus, certains utilisateurs peuvent souffrir de déficiences, comme le daltonisme, qui affectent la perception des couleurs. C'est pourquoi dans la plupart des systèmes, les échelles de couleurs sont configurables et un panel d'échelles différentes est souvent proposé. Différents outils comme PravdaColor [140] ou ColorBrewer [36], qui ont plus récemment été généralisés [255], permettent de générer et configurer des échelles de couleur perceptuellement efficaces.

Pour des valeurs quantitatives, la distribution des valeurs peut influer sur l'efficacité de l'échelle de couleur. En effet, si l'échelle de couleur est uniforme mais pas la distribution des valeurs, alors les intervalles de valeurs très denses ont autant de couleurs pour représenter leurs valeurs que les intervalles avec très peu de valeurs. Aussi toutes les valeurs des intervalles denses ne sont pas visibles. C'est pourquoi il faut pouvoir adapter l'échelle de couleur, soit en modifiant les bornes de l'intervalle de valeurs, en changeant le nombre de points de contrôles c'est-à-dire les points à partir desquelles la teinte de la couleur change, ou bien en utilisant une fonction de transfert non linéaire. Des échelles de couleur qui s'adapter à la distribution ont été développées [195] afin d'adapter l'échelle à l'histogramme des valeurs ou au Box-Whisker plot (Figure 3.3).

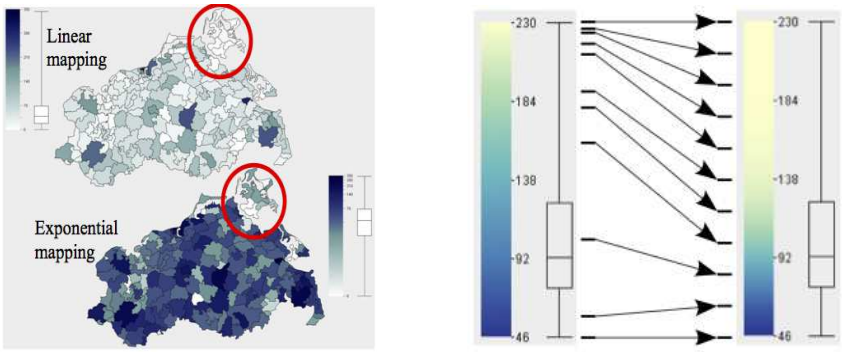


FIGURE 3.3 – Exemple d'échelle de couleur auto-adaptative [195].

3.2.2 Application à la coloration des proximités

Dans la visualisation interactive des proximités, la distribution des proximités n'est en générale pas uniforme d'autant plus lorsque des points atypiques sont présents dans les données. Aussi il apparait intéressant d'adapter l'échelle de couleur, par égalisation d'histogramme par exemple, pour qu'elle mette correctement en valeur la distribution des proximités, en attribuant plus de couleurs aux intervalles de valeurs les plus denses. Cependant la distribution réellement visualisée correspond à un vecteur de la matrice et non à toutes les proximités de la matrice. Aussi il faudrait adapter l'échelle de couleur dynamiquement lorsque qu'un nouveau point de référence est sélectionné par l'utilisateur.

Mais dans ce cas de figure, on ne peut plus comparer équitablement les différentes visualisations des proximités d'un point de référence à un autre. Or si on ajuste l'échelle de couleur à la totalité des proximités, alors l'échelle ne sera pas correctement distribuée pour des points de référence atypiques et la visualisation des proximités ne sera alors pas équitablement comparable avec celle des autres références. Aussi nous avons choisi de ne pas adapter l'échelle de couleur à la distribution des proximités.

En revanche on peut afficher dynamiquement les paramètres de la distribution du vecteur des proximités par le biais d'un histogramme ou d'un Box-Whisker plot à coté de l'échelle de couleur. On peut également transformer la matrice des proximités, par l'utilisation d'un log par exemple, pour corriger des distributions trop déséquilibrées. Mais le déséquilibre de la distribution des proximités peut aussi être vu comme une information en tant que tel de la singularité des données et de la mesure de similarité utilisée.

L'échelle de dégradé de gris initialement utilisée est pertinente pour montrer les écarts entre proximités mais elle peut sembler assez austère pour des non-experts. Or l'échelle contribue à l'image que l'on se fait d'une visualisation et elle doit donc être relativement attractive pour susciter l'envie d'interagir [158]. C'est pourquoi nous avons envisagé d'autres nuances de couleurs.

Nous nous sommes intéressé à l'échelle introduite par Tominski et al. [227] pour des tâches de comparaison entre régions (Figure 3.4). Cette échelle commence par une couleur blanche indiquant la distance nulle, puis continue avec des nuances de violet et de vert, avant de terminer par du noir indiquant la distance maximum dans la matrice de similarité. Cette échelle est segmentée en différentes teintes bien séparées. Elle permet ainsi de discriminer visuellement des intervalles de proximité ou des frontières entre zones de même proximité, c'est-à-dire des frontières entre clusters. On peut en effet observer les premiers sauts dans la distribution des proximités, qui indiquent la limite entre les proximités intra-cluster et les proximités inter-cluster, dans le cas où les données correspondent à un modèle de mixture de Gaussiennes. Pour des données où il n'y a pas de séparation nette entre les clusters, mais des structures topologiques plus complexes, cette échelle de couleur atteint ses limites et on pourra préférer un dégradé perceptuellement uniforme de bleu par exemple.

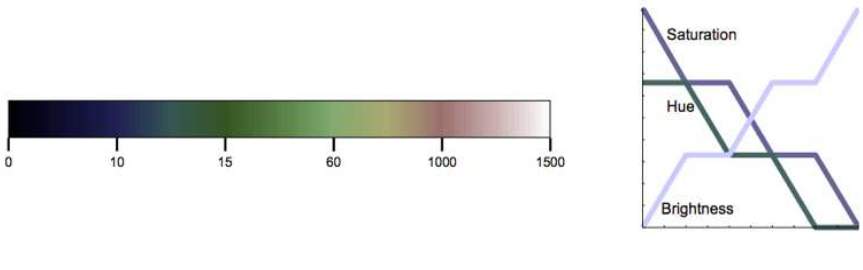


FIGURE 3.4 – Echelle de couleur pour des tâches de comparaison [227].

3.2.3 Support de l'encodage couleur

Les cellules de Voronoï étant de tailles arbitraires, ceci peut introduire un biais dans l'analyse de la visualisation des proximités. En effet, la taille d'une cellule dépend de l'espacement entre les points, aussi de larges cellules indiquent une région relativement peu dense de la projection. Mais cette taille n'est pas directement corrélée avec les couleurs affichées qui représentent les proximités à un point donné dans l'espace des données. Aussi parce que les proximités encodées par la couleur sont relatives au point de référence et non au point générant la cellule de Voronoï, alors ce support peut prêter à confusion et générer un biais de perception. C'est pourquoi nous avons envisagé d'autres supports d'application de la couleur comme les points ou l'arrière plan en utilisant une interpolation de la couleur (Figure 3.5).

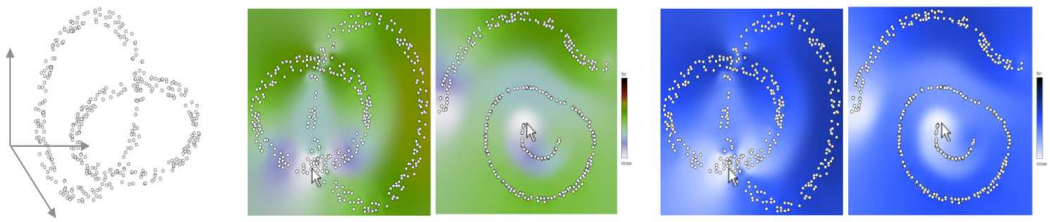


FIGURE 3.5 – ProxiViz avec une coloration interpolée et utilisé sur le jeu de données des anneaux entrelacés en 3D (à gauche), avec l'échelle de couleur de Tominski et al. [227] (au centre) et une échelle en dégradé de bleu (à droite). La première échelle de couleur part du blanc pour une distance nulle en passant par du bleu, violet, puis vert jusqu'au noir pour une distance maximum. Cette échelle est plus segmentée que l'échelle de bleu et peut ainsi permettre de mieux distinguer des clusters, lorsque la distribution des proximités correspond à la distribution des changements de couleur.

Comme la couleur des points est souvent utilisée pour représenter les étiquettes de classe, nous avons choisi d'utiliser une coloration en arrière plan avec une interpolation de Shepard [203] entre les points qui permet obtenir un lissage de la représentation que l'on avait avec les cellules de Voronoï. Cette interpolation calcule la couleur $u(x)$ au pixel x en utilisant la pondération inverse des distances de la couleur u_i en chaque point i de la projection :

$$u(x) = \sum_{i=0}^{N} \frac{w_i(x)u_i}{\sum_{j=0}^{N} w_j(x)} \text{ , avec } w_i(x) = \frac{1}{||x - x_i||^k}$$

On choisit un facteur de voisinage $k = 2$ pour préserver au mieux l'information locale, un facteur plus élevé fait tendre l'interpolation vers une mosaïque de Voronoï. Cette interpolation permet de donner les tendances globales des distances par lissage tout en préservant l'information locale avec un cercle de dégradé autour des points affichant leur "vraie" couleur.

Nous avons implémenté l'interpolation de Shepard [204] pour la coloration à l'aide d'un programme Shader, d'abord en `DirectX` avec `C\#`, puis en `WebGL` avec `d3.js` [32]. D'autres techniques d'interpolation comme l'interpolation de Sibson [208] peuvent également être envisagées mais sont plus complexes à implémenter [167].

3.3 Interaction de navigation

Dans la version originale de la visualisation interactive des proximités, le point de référence était sélectionné à la souris par un clic sur le point. Pour fluidifier l'interaction de navigation, nous proposons de sélectionner la référence lors du survol de la souris, c'est-à-dire en sélectionnant le point le plus proche du curseur de la souris. Pour se faire, on peut utiliser les cellules de Voronoï comme support de l'interaction de navigation et le passage du curseur d'une cellule à l'autre, afin de détecter quand sélectionner un nouveau point comme référence. Une fois un nouveau point sélectionné, on affiche immédiatement les couleurs correspondant au nouveau vecteur de la matrice de similarité. On peut éventuellement utiliser une animation de transition entre les couleurs.

Toutefois cette approche de sélection nécessite des précautions, car en explorant la projection à l'aide du curseur de la souris, la distribution des couleurs peut changer du tout au tout en passant d'un point à l'autre. En effet, à cause des artefacts de faux voisinages, on ne peut pas garantir que des points voisins au sens de Delaunay, c'est-à-dire voisins par leurs cellules de Voronoï sur la projection, soient en réalité voisins dans l'espace des données. Ces artefacts créent des effets de clignotement qui peuvent perturber les utilisateurs lors de leur exploration (Figure 3.1).

Pour résoudre ce problème d'interaction, il nous faut distinguer les points autour de la référence en fonction de leur proximité d'origine à celle-ci. Nous pourrions modifier l'espace moteur de la souris afin de parvenir à ces fins, à la manière du pointage sémantique [27]. Mais pour des raisons d'implémentation, nous proposons une solution plus simple basée sur un délai dans le déclenchement de la sélection, afin d'éviter ces changements brutaux de couleur. Nous proposons de configurer ce délai en fonction de la proximité à la référence courante : $T = T_1 \times d(x_i, x_j) + T_0$, avec T_0 une constante de temps fixée par exemple à 100ms et T_1 un facteur d'échelle dépendant de la taille de la projection. A chaque fois qu'un nouveau point doit être sélectionné, le nouveau délai annule le précédent. On peut ainsi sélectionner les points les plus proches de la référence courante dans l'espace des données sans sélectionner des faux voisinages. En effet, les points les plus proches seront sélectionnés instantanément (en T_0 temps) alors que les plus éloignés présenterons un délai d'attente avant d'être sélectionnés. Tout dépend ensuite de la façon dont l'utilisateur explore la projection (Figure 3.6).

On peut distinguer deux stratégies :

- Exploration libre : l'utilisateur veut explorer toute la projection, sans stratégie précise, il peut donc être intéressant de découvrir des artefacts.
- Exploration d'un cluster : l'utilisateur veut explorer un cluster spécifique, il souhaite donc se déplacer de points voisins en points voisins dans l'espace des données. Il faut donc éviter la sélection de faux voisinages et permettre de visiter des composantes non connexes du même cluster sur la projection (des déchirures). Dans ce cas de figure, le délai permet d'éviter la sélection de faux voisinages (si l'utilisateur sélectionne les points en un temps autour T_0) et si il déplace assez rapidement la souris d'une composante à l'autre.

Le système de délai est assez difficile à paramétrer, car il doit permettre de réduire les effets de clignotement sans pour autant nuire au confort de la navigation en saccadant la sélection pour des individus voisins dans l'espace des données. Mais il permet de résoudre simplement les problèmes de clignotement dûs aux artefacts de faux voisinages. Il faut garder à l'esprit que la navigation est fluide si il y a une corrélation très forte entre la position des points et la coloration de la vi-

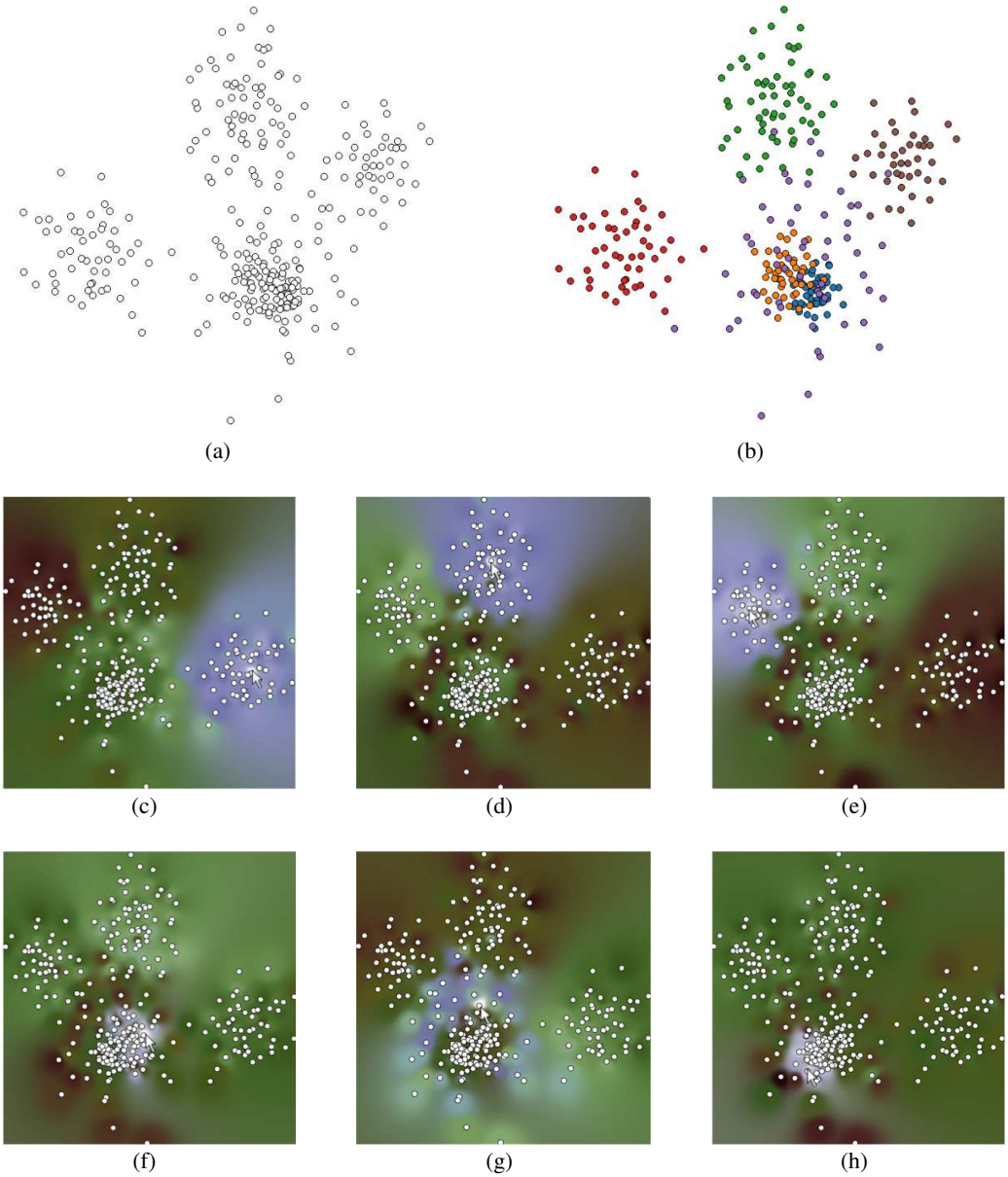


FIGURE 3.6 – Illustration de ProxiViz sur le jeu de données fil-rouge. La projection sans ajout d'information ne rèvele que 4 clusters (a). ProxiViz permet de réveler les 6 clusters (c, d, e, f, g, h) correspondant aux différentes classes (b) présentes dans les données.

sualisation des proximités, c'est-à-dire si la projection présente plutôt des artefacts de déchirure que de faux voisinages. De plus les artefacts topologiques interviennent à différents niveaux de granularité, selon que l'on considère des clusters ou que l'on se place au niveau des relations entre points. Le mécanisme de sélection peut également être activé/inhibé comme un *mode d'interaction*, afin de permettre de passer d'un bout à l'autre de la projection sans changer les couleurs de la visualisation des proximités. D'autres solutions d'interaction sont proposées dans le chapitre 6.

3.4 Discussion

Différents inconvénients peuvent être trouvés dans la représentation des proximités par le biais de la couleur, notamment par rapport au passage à l'échelle, comme le contraste des couleurs par rapport à la distribution des proximités ou les problèmes d'occlusion qui cachent la coloration. Les sauts dans l'échelle de couleur, matérialisant des proximités inter-clusters, ne sont pas toujours très visibles et ils dépendent de la distribution des proximités. Aussi une proximité intra-cluster n'est pas obligatoirement encodée par une couleur claire ce qui n'est pas toujours intuitive pour des non-experts qui ne pensent pas en termes de densité dans un cluster.

L'absence de guidage dans l'exploration des proximités oblige à naviguer sur toute la projection pour en découvrir toutes les subtilités, ce qui peut prendre du temps. De plus, étant relatif à un unique point, la généralisation des motifs colorés encodant les proximités à des clusters n'est pas triviale. D'autant que les artefacts topologiques peuvent compliquer la tâche d'analyse visuelle de ces motifs : les faux voisinages bruitent la coloration autour de la référence et les déchirures impliquent d'explorer des régions éloignés sur la projection. Ces problèmes sont expliqués plus en détails dans le chapitre 6. Au delà de ces problématiques, ProxiViz est une technique relativement simple qui repose uniquement sur un encodage couleur des proximités d'origine relative à un individu. Elle ne nécessite donc pas une longue période d'apprentissage et peut donc être évaluée quantitativement par des utilisateurs non expérimentés.

3.5 Conclusion

Dans ce chapitre, nous avons revisité la visualisation interactive des proximités afin d'améliorer son encodage visuel et son interaction de navigation. Le choix d'une "bonne" échelle de couleur est complexe de même que le support d'application de la coloration sur la projection. Nous avons proposé un nouvel encodage couleur, basé sur une interpolation de la couleur entre les points, plus homogène que les cellules de Voronoï. Nous avons également discuté les problèmes de navigation liés à cette technique, que nous appellerons désormais ProxiViz. Le chapitre suivant introduit une évaluation quantitative de l'efficacité de ProxiViz selon différentes variantes d'encodage de la couleur et par rapport à différentes techniques de l'état de l'art.

4

Evaluation de l'encodage visuel de ProxiViz

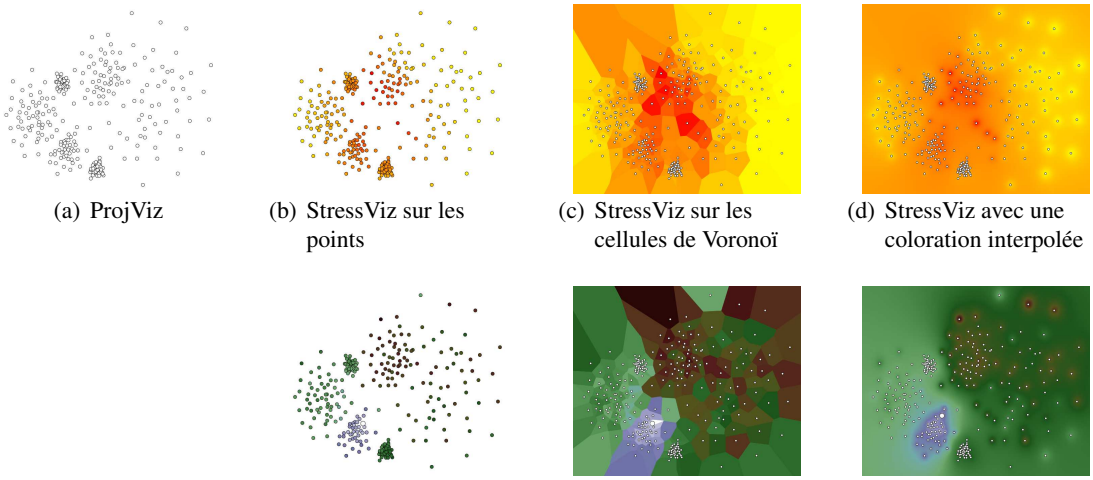
ProxiViz permet de révéler les artefacts de projection mais son efficacité pour aider l'analyse visuelle des projections n'a pas été vérifiée. Le stress local de la projection quantifie en chaque point la distorsion des distances 2D par rapport aux proximités d'origine. Cette information affichée statiquement sur la projection, par le biais d'une coloration, permet d'indiquer les zones de distorsions. A ce jour, aucune étude ne s'intéresse à la façon dont les utilisateurs prennent en compte ces retours sur la qualité de la projection dans leurs inférences sur l'espace des données. Ce chapitre présente une expérience contrôlée comparant ProxiViz avec la projection classique sans ajout d'information et une projection colorée en fonction de l'information de stress. Plusieurs encodages couleurs sont pris en compte afin d'obtenir des retours quantitatifs et qualitatifs permettant d'améliorer ProxiViz. Nous considérons une tâche d'énumération de clusters et évaluons les performances de chaque technique et encodage selon deux types de projections : celles favorisant des artefacts de faux voisinages et celles introduisant des déchirures.

4.1 Motivation

Dans un cadre d'analyse exploratoire de données de grande dimension, la projection est le plus souvent visualisée sous forme d'un nuage de points sans coloration. L'information de stress quantifie les distorsions des distances 2D par rapport aux proximités d'origine. L'affichage statique cette information directement sur la projection permet d'indiquer quels points sont "mal placés", par le biais d'un encodage couleur du stress local. Ce retour sur la qualité de la projection permet ainsi d'estimer la fiabilité de l'analyse visuelle selon les différentes régions de la projection. ProxiViz permet de révéler interactivement les artefacts de projection relatifs à un point de référence, par le biais d'un encodage couleur des proximités d'origine.

Peu d'études s'intéressent à la façon dont les utilisateurs prennent en compte ces informations additionnelles sur les artefacts dans leurs inférences sur l'espace des données issues de l'analyse visuelle de la projection. Nous avons donc réalisé une expérience contrôlée afin de déterminer si l'ajout des informations de stress et de proximité permet d'améliorer la précision de l'analyse visuelle d'une projection. Dans la suite on note la visualisation du stress *StressViz* et la visualisation des proximités *ProxiViz*. On nommera la projection sans ajout d'information *ProjViz*.

L'encodage visuel utilisé par StressViz et ProxiViz pour afficher les informations additionnelles peut influer sur l'efficacité de chaque technique (précision et temps de réponse). Comme décrit précédemment, plusieurs encodages couleurs sont possibles selon le support d'application de la couleur (Figure 4.1). Nous avons considéré trois encodages couleurs : coloration de chaque point, de chaque cellule de Voronoï, interpolation en arrière plan de la couleur entre chaque point.



(a) ProjViz (b) StressViz sur les points (c) StressViz sur les cellules de Voronoï (d) StressViz avec une coloration interpolée

(e) ProxiViz sur les points (f) ProxiViz sur les cellules de Voronoï (g) ProxiViz avec une coloration interpolée

FIGURE 4.1 – Résumé des différents encodages couleurs utilisés pour StressViz et ProxiViz : coloration des points, coloration des cellules de Voronoï et interpolation de la coloration. Pour StressViz, l'échelle de couleur représente le maximum de stress par la couleur rouge et le minimum par la couleur jaune. Pour ProxiViz, l'échelle de couleur a été décrite dans le chapitre précédent.

Pour des raisons pratiques liées au contrôle de la durée de l'expérimentation, nous considérons exclusivement une tâche de clustering visuel. Cette tâche consiste à estimer le nombre de clusters dans l'espace des données à partir des clusters 2D visibles sur la projection. Le chapitre suivant présente une seconde évaluation de ProxiViz sur un panel plus large de tâches d'analyse visuelle. La précision de cette tâche dépend du type d'artefacts présents sur la projection. En effet des faux voisinages peuvent joindre des clusters séparés dans les données et les déchirures peuvent les séparer en différentes composantes non connexes sur la projection (Figure 2.12). Nous avons donc considéré deux types de projections, chacune favorisant un type d'artefact, afin d'estimer si cette taxonomie des artefacts impacte significativement la précision du clustering visuel.

Dans ce chapitre, nous décrivons l'expérience contrôlée que nous avons réalisé afin de répondre aux questions suivantes :

Q1. ProxiViz et StressViz sont-ils plus précis que ProjViz pour une tâche de clustering visuel ?

Q2. L'encodage couleur impacte-t-il la précision de StressViz et ProxiViz ? C'est à dire un encodage couleur est-il meilleur qu'un autre ?

Q3. L'interactivité de ProxiViz impacte-t-elle fortement les temps de réponse de l'analyse visuelle par rapport ceux de ProjViz ?

Q4. Le type d'artefacts topologiques présents impacte-t-il la précision du clustering visuel ?

4.2 Etudes utilisateurs des projections

Différents types d'études existent pour comparer les performances des algorithmes de projection [26]. Ces évaluations considèrent des mesures de qualité et un paramétrage des algorithmes pour les comparer sur des données benchmark. Cependant ces études ne s'intéressent pas aux enjeux d'interprétation des projections par des utilisateurs.

Peu d'études utilisateurs ont été réalisées sur la façon dont les projections sont interprétées [115]. Une expérience récente évalue comment les experts et novices jugent la qualité d'un nuage de points [126], montrant sans trop de surprise que les experts sont plus précis dans leurs analyses que des novices qui font plus d'erreurs d'interprétation. Une étude récente a comparé l'analyse visuelle du nuage de points 2D et 3D, tels que des projections, avec les matrices de nuages de points (SPLOM) [197]. Les résultats montrent que les matrices de nuages de points 2D sont les plus efficaces pour réellement aider les utilisateurs dans leur exploration visuelle des données.

Des études ont également comparé différents paysages d'information (information landscapes) avec les nuages de points 2D sur des données spatiales et non spatiales [229] [230]. En considérant des tâches non triviales d'identification d'intervalle de valeur et de mémorisation de position, avec des échelles de couleurs basés soit sur des nuances de teinte soit sur des contrastes de gris, ces évaluations révèlent que l'affichage basé sur les points est plus efficace que les paysages d'information et que une échelle de couleur basée sur la teinte est plus efficace qu'une échelle de gris. Cependant ces résultats ne sont pas directement applicables à des tâches de plus haut niveau nécessitant des inférences à partir des valeurs encodées par le biais de la coloration.

Des évaluations de l'efficacité des visualisations basées sur les pixels, avec des tâches utilisant la couleur comme encodage visuel [30], ont montré que la précision dépend de la charge cognitive

de la tâche et que pour des tâches de haut niveau comme l'évaluation de changements d'erreurs, la précision est liée directement à la distance entre les régions à comparer dans l'espace des couleurs.

Parmi ces évaluations considérant les projections de données, aucune ne s'intéresse à la façon dont les utilisateurs appréhendent la qualité de la projection. En particulier aucune évaluation ne cherche à déterminer dans quelle mesure les artefacts de projection impactent l'analyse visuelle de projections. Aussi nous proposons d'apporter des éléments de réponse à cette question, avec l'expérience controlée décrite dans la section suivante.

4.3 Expérience contrôlée

Dans cette section, nous décrivons d'abord les différents facteurs considérés dans l'expérience, puis sa conception générale et son déroulement, avant d'aborder les résultats.

4.3.1 Techniques

Nous considérons les techniques suivantes :

ProjViz est statique et affiche seulement les proximités 2D approximées à partir des proximité d'origine. Elle affiche un point pour chaque individu sans aucun encodage couleur (monochrome).

StressViz est statique et affiche les informations relatives au stress de la projection. On affiche pour chaque point sa mesure de stress local obtenue à partir du critère d'optimisation de l'algorithme utilisé pour générer la projection. On combine cette visualisation avec les 3 encodages couleurs décrits précédemment. Pour cette technique nous proposons d'utiliser une échelle de couleur du jaune au rouge. Cette échelle est jugée comme culturellement intuitive [21], car la couleur rouge indique le maximum de stress, c'est-à-dire le maximum de risque d'effectuer des inférences erronées sur l'espace des données. De même la couleur jaune indique le minimum de stress, c'est-à-dire le minimum de risque de mal interpréter la position des points.

ProxiViz est interactive et affiche les informations de proximité d'origine relatives à un individu référence. Quand la souris bouge au dessus de la visualisation, le point le plus proche est sélectionné interactivement comme *référence* [14] (l'activation de la sélection automatique peut être contrôlée par un clic ou une touche du clavier) ; L'encodage couleur de la visualisation est immédiatement mis à jour pour afficher sur chaque point sa proximité à la référence dans l'espace des données. On combine cette visualisation interactive avec les 3 encodages couleur précédemment identifiés. Nous considérons l'échelle de couleur introduite par Tominski et al. [227] décrite précédemment. Cette échelle de couleur permet de mieux différencier les frontières entre clusters. Elle commence avec une couleur blanche indiquant la distance nulle et continue avec des nuances de violet puis de vert pour finir par du noir indiquant le maximum de distance par rapport à la référence.

De plus, le choix de 2 échelles de couleur différentes permet d'éviter que les utilisateurs confondent les conditions de chaque technique. Dans notre configuration, chaque échelle de couleur est clairement reliée à une information différente à interpréter : le stress ou la proximité dans l'espace des données.

Les cellules de Voronoi sont pré-calculées pour chaque projection en utilisant l'algorithme de Fortune [86]. Pour l'interpolation de la couleur, on utilise l'interpolation de Shepard [203] entre les points. Cette interpolation utilisée pour ProxiViz affiche les tendances globales des proximités à la référence dans chaque zone de la projection, tout en préservant une information locale avec un cercle de dégradé de couleur autour de chaque point. En combinant les 3 techniques avec les 3 encodages couleurs, on obtient 7 valeurs pour le facteur Technique, nommés dans la suite comme suit (Tableau 4.1) : PROJVIZ, STRESS$_P$, STRESS$_V$, STRESS$_I$, PROXI$_P$, PROXI$_V$, PROXI$_I$.

Coloration	**Information**		
	Aucun	Stress	Proximité
Aucun	PROJVIZ (1)		
Points		STRESS$_P$ (2)	PROXI$_P$ (5)
Voronoï		STRESS$_V$ (3)	PROXI$_V$ (6)
Interpolation		STRESS$_I$ (4)	PROXI$_I$ (7)

TABLE 4.1 – Tableau résumant les différentes techniques considérées.

4.3.2 Jeux de données

Nous avons étudié 34 jeux de données réelles fréquemment utilisés en analyse de données [87]. Les jeux de données ont été choisis de manière à varier leurs caractéristiques : type (signal, images), cardinalité, dimensionnalité, quantité de clusters, etc.. Pour des données de taille trop importante, des sous échantillons d'individus et de dimensions ont été crées de manière à obtenir environ 500 points et moins de 1000 dimensions par jeu de données. Cette contrainte permet d'éviter les problèmes liés au passage à l'échelle du calcul de la matrice de similarité, ainsi que de limiter les phénomènes d'occlusions et de ralentissement dans les interactions. Nous avons considéré comme mesure de similarité une distance Euclidienne sur l'ensemble des dimensions. Les jeux de données ont ensuite été filtrés en fonction de la pureté des classes, c'est-à-dire faible chevauchement des clusters sous-jacents aux classes, afin que la structure sous-jacente aux classes reflète le modèle des classes.

Finalement, nous considérons pour cette expérience 3 jeux de données réelles :

Letter Recognition Dataset (*letters* de [87]) : Ce jeu de données se compose d'images de lettres dans 20 polices différentes. Les images sont distordues pour créer différents individus représentatifs d'une même lettre. Nous avons utilisé un échantillon de ce jeu de données composé de 5 lettres (A, E, I, O, U) avec 50 individus choisis aléatoirement pour chacune des lettres. Chaque individu se compose de 16 caractéristiques extraites automatiquement à partir des images d'origine.

CMU Face Images Dataset (*faces* de [87]) : Ce jeu de données ce compose d'images en niveaux de gris de visages de personnes prises en photo. Nous avons utilisé un échantillon aléatoire de différentes personnes (an2i / at33 / saavik / steffi) prises en photos dans différentes positions (left / right / straight / up), avec différentes expressions (angry / happy / neutral / sad) et avec/sanc lunettes de soleil, soit 128 images au total. Chaque individu se compose de 32x30 pixels, soit 960 dimensions.

Teapot Dataset (*teapot* de [274]) : Ce jeu de données se compose de 365 images en niveaux de gris d'une théière prise en photos à 360 degrés. Le jeu de données original contenait 400 images RGB de 101x76 pixels. Il a ensuite été adapté pour composer deux variétés disjointes : une correspondant à la théière avec son anse à gauche et l'autre avec l'anse à droite. Chaque individu se compose de 16x12 pixels, soit 192 dimensions.

Nous avons également généré 3 jeux de données synthétiques en utilisant la procédure suivante : k centroids de dimension d sont tiré aléatoirement en suivant une loi Normale en d-dimensions. Chaque centroid est ensuite utilisé comme moyenne d'une loi Normale, où chaque écart type est calculé comme la moitié de la distance minimum entre le centroid considéré et les autres centroids. Cette contrainte permet de réduire le chevauchement entre les clusters dans l'espace des données. n individus synthétiques sont finalement tirés aléatoirement à partir de chacune des distributions construites précédemment. Différents jeux de données, avec des nombres d'individus aléatoires, ont été générés à partir de cette procédure en utilisant 5, 10 et 200 dimensions. Parmi les jeux de données générés, nous avons conservé un jeu de 141 individus en 5 dimensions avec 3 clusters, un autre de 293 individus en 10 dimensions avec 6 clusters et un dernier de 361 individus en 200 dimensions avec 7 clusters (Tableau 4.2).

Nom	Dimensions	Individus	Clusters
synthetic1	5	141	3
synthetic2	10	293	6
synthetic3	200	361	7
letters	16	250	5
faces	960	128	4
teapot	192	365	2

TABLE 4.2 – Tableau résumant les jeux de données utilisés dans l'expérience.

4.3.3 Type d'artefacts de projection

Pour évaluer l'impact du type d'artefact sur le clustering visuel de la projection, nous utilisons des projections composées principalement de *déchirures* et de *faux voisinages*. Ces projections ont été générées en utilisant l'algorithme Local Multidimensional Scaling [49] (LMDS). Ce dernier permet de paramétrer la fonction objectif pour correspondre soit à une projection non-linéaire de Sammon [186], soit à une Analyse en Composantes Curvilignes [66]. La première fonction objectif pénalise les déchirures et favorise les faux voisinages alors que la seconde a l'effet inverse. Le paramètre λ de l'algorithme LMDS permet donc de définir un équilibre entre déchirures et faux voisinages, comme le montre sa fonction objectif :

$$E = \frac{1}{2}\sum_{i}\sum_{j\neq i}(d_{i,j} - d^*_{i,j})^2[\lambda F_\sigma(d^*_{i,j}) + (1-\lambda)F_\sigma(d_{i,j})]$$

Avec F_σ une fonction de Heaviside : $F_\sigma(x) = 1, \text{si } x \leq \sigma$ et $F_\sigma(x) = 0$ sinon. Le paramètre de voisinage σ tend vers zéro durant l'optimisation qui consiste en une descente stochastique de gradient. Pour définir la mesure de stress représentée par StressViz, on utilise le résultat de la fonction objectif avec $\sigma^* = min(d^*_{i,j})$.

Lorsque $\lambda = 1$, pour une distance dans les données $d^*_{i,j}$ suffisamment grande telle que $F_\sigma(d^*_{i,j}) = 0$, alors l'écart entre la distance 2D $d_{i,j}$ et $d^*_{i,j}$ n'influencera pas la fonction de stress. Ceci implique que des points i et j éloignés dans l'espace des données peuvent être proches dans la projection, autrement dit l'algorithme favorise la création de faux voisinages. Réciproquement, lorsque $\lambda = 0$, l'algorithme favorise des déchirures. Pour contrôler le type d'artefacts ont utilise donc les deux configurations extrêmes de l'algorithme LMDS ($\lambda = 0$ et $\lambda = 1$).

4.3.4 Tâches

Pour cette expérience contrôlée, on se concentre sur une tâche de clustering visuel. L'identification de clusters visuellement est une tâche de haut-niveau qui nécessite de l'inférence à partir de la position relative des points et des motifs qui émergent du nuage de points. En utilisant ProxiViz, l'inférence n'est plus basée seulement sur les points mais également sur la coloration. L'inférence à partir de la couleur n'est pas directe car ce sont les distances relatives entre elles qui permettent de distinguer des clusters dans l'espace des données. En utilisant StressViz, l'inférence n'est pas aussi facile et elle dépend à la fois de la position des points et de la coloration.

Un premier pilote de l'expérience a été effectué avec une tâche d'identification utilisant une technique de lasso pour que l'utilisateur dessine les frontières de chaque cluster. Ce pilote a montré que cette tâche n'était pas appropriée pour une évaluation car les utilisateurs passaient trop de temps à dessiner précisément les frontières avec le lasso. C'est pourquoi nous avons décidé de remplacer la tâche d'identification visuelle des clusters (clustering visuel) par une tâche plus simple d'énumération du nombre de clusters. Les participants devaient répondre à la question suivante : "combien de clusters pouvez vous identifier ?". Ils devaient ensuite valider à l'aide de la souris une des réponses présentes dans un formulaire (allant de 1 à 8 clusters). Ils devaient considérer des clusters contenant au moins 10 points (en agrégeant des sous-clusters le cas échéant). La réponse 1 cluster signifiait qu'ils n'étaient pas en mesure de distinguer des clusters dans les données à partir de la projection et de la technique utilisée.

4.3.5 Conception de l'expérience

Nous considérons une expérience contrôlée avec des mesures répétées (repeated-measure) avec les facteurs intra-sujets (within-subjects) suivants : Technique (PROJVIZ, STRESS_P, STRESS_V, STRESS_I, PROXI_P, PROXI_V, PROXI_I), Type d'artefacts (Déchirures, Faux voisinages). En résumé, l'expérience comprend :

6	jeux de données	$\times$
2	types d'artefacts	=
12	tests	$\times$
7	techniques (blocs)	=
84	tests par participants	$\times$
21	participants	=
1,764	**tests au total**	

On mesure le nombre d'erreurs et les temps de réponse. Le temps de réponse (c'est-à-dire le temps pour valider une réponse) est enregistré en secondes. L'erreur est calculée comme la différence signée entre le nombre de clusters trouvés $p \in [1,8]$ et le nombre de clusters réellement présents dans l'espace des données $p^* \in [2,7]$. On utilise dans la suite un pourcentage de précision basé sur cette erreur :

$$v = \begin{cases} |\frac{p-p^*}{8-p^*}| \times 100, \text{ si } p > p^* \\ |\frac{p-p^*}{1-p^*}| \times 100, \text{ si } p \leq p^* \end{cases} \tag{4.1}$$

4.3.6 Participants et procédure

21 participants (14 hommes et 7 femmes), de 21 à 35 ans (28 ans en moyenne), de deux laboratoires spécialisés en analyse de données, ont complété l'évaluation. Tous les participants, dont 6 étudiants, étaient informaticiens. Ils avaient tous une vue normale sans problèmes de vision des couleurs. Seuls 5 participants n'avaient jamais utilisé de projection de données avant et tous les autres en utilisaient au moins une fois par mois ou plus.

Le système d'évaluation à été implémenté en `C\#` avec un programme `Shader DirectX` pour l'interpolation de la couleur. Les participants étaient assis à une distance d'environ 50cm devant un écran HP Compaq LCD 22 pouces avec une résolution de 1680x1050 pixels. Les participants ont répondu à chaque question en sélectionnant à la souris une des réponses proposées dans le formulaire et en cliquant sur un bouton de validation.

Le temps de réponse est mesuré une fois que la projection apparaît à l'écran et s'arrête quand le participant à validé une réponse. Après avoir passé toutes les questions, les participants ont rempli un questionnaire de manière à recueillir des informations démographiques (âge, statut, etc.) et des retours de préférences subjective sur chaque technique. Des notes ont également été prises pour suivre le comportement des utilisateurs avec chaque technique.

Chaque participant a été confronté successivement à 7 blocs (un pour chaque technique) de 12 tests (6 jeux de données projetés avec les deux configurations de l'algorithme de projection). L'ordre des blocs et des tests est contre-balancé (counterbalanced). Pour éviter l'effet d'apprentissage, les 6 jeux de données ont été re-projetés avec les 2 configurations de l'algorithme de projection pour chaque bloc ; la re-projection produit des résultats légèrement différents, en particulier en termes d'orientation et de symétrie.

Entre chaque bloc, les participants pouvaient faire une pause. Ils pouvaient également s'arrêter pendant une durée plus longue entre chaque test. Une durée de 30-60 secondes maximum pour chaque test devait être respectée, pour une durée totale d'évaluation d'environ 40 minutes, précédées par 20 minutes de présentation de l'évaluation et un entraînement sur 7 jeux de données synthétiques, un pour chaque technique, pour apprendre comment utiliser et interpréter les techniques.

4.3.7 Hypothèses

StressViz indique où les points sont “mal placés” sur la projection, mais cette technique ne donne pas d’informations sur la position "idéale" de ces points. On peut donc supposer que StressViz n’apporte pas une aide importante au processus d’analyse visuelle de la projection. En revanche, ProxiViz permet non seulement de révéler les points étant des artefacts topologiques, mais également de déterminer interactivement la position "idéale" de ces artefacts, c’est-à-dire de visualiser à proximité de quels autres points ils devraient être positionnés. On peut donc supposer que ProxiViz apporte une aide significative au processus d’analyse visuelle de la projection.

Nos hypothèses sont les suivantes :

H1 On s’attend à ce que les participants soient plus précis avec l’ajout d’informations de proximité que d’informations de stress. On suppose également que les participants seront moins précis sans ajout d’informations relatives aux artefacts. On suppose donc que les participants seront significativement plus précis avec les variantes de ProxiViz (PROXI$_P$, PROXI$_V$, PROXI$_I$) qu’avec les variantes de StressViz (STRESS$_P$, STRESS$_V$, STRESS$_I$), lesquelles permettront d’être significativement plus précises qu’avec PROJVIZ.

H2a Aucune différence significative en termes de précision et de temps de réponse ne sera trouvée entre les variantes de ProxiViz (PROXI$_P$, PROXI$_V$, PROXI$_I$), car la couleur fournie la même information indépendamment de l’encodage.

H2b Aucune différence significative en précisions et en temps de réponse ne sera trouvée entre les variantes de StressViz (STRESS$_P$, STRESS$_V$, STRESS$_I$), car la couleur fournie la même information indépendamment de l’encodage.

H3 Les utilisateurs seront significativement plus rapide en utilisant PROJVIZ qu’avec les autres techniques. Les participants seront significativement plus lents avec les variantes de ProxiViz (PROXI$_P$, PROXI$_V$, PROXI$_I$) que les variantes de StressViz (STRESS$_P$, STRESS$_V$, STRESS$_I$) et PROJVIZ, car l’interaction nécessite plus de temps que celui nécessaire aux inférences à partir d’une visualisation statique.

H4 Les utilisateurs seront significativement plus précis sur des projections favorisant des déchirures plutôt que des faux voisinages.

4.4 Résultats

Cette section décrit les résultats statistiques en termes de précision et de temps de réponse (Figure 4.2). Comme nous considérons des facteurs intra-sujets avec répétition des mesures et que pour toutes les tâches la précision ne suit pas une distribution de loi Normale, nous utilisons le test non-paramétrique de Friedman pour analyser la variance ainsi que des tests pairés de Wilcoxon (de rang signé) pour analyser 2 distributions indépendantes non-paramétriques. Le temps de réponse est log-transformé pour mieux correspondre à une distribution de loi Normale. On utilise l’ANOVA pour analyser la variance et des t-tests pairés pour comparer les temps de réponse.

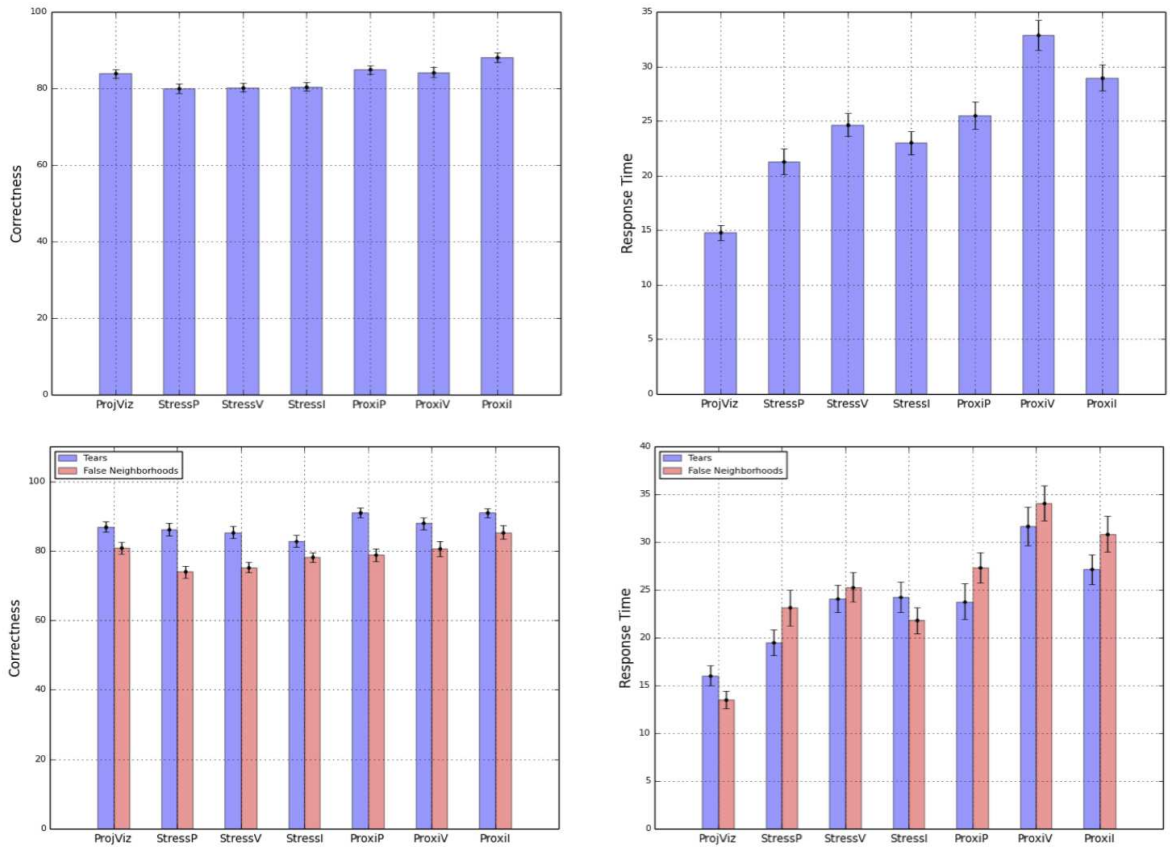


FIGURE 4.2 – Résultats en termes de moyennes et écarts types de précision (en %) et de temps de réponse (en secondes) pour chaque technique (au dessus) et pour chaque technique par rapport à chaque type d'artefact (en dessous, faux voisinages en rouge et déchirures en bleu).

Précision

En termes de précision, le test de Friedman révèle un effet significatif du facteur Technique ($\chi^2(6) = 55.9, p < .0001$). Les participants ont été précis à 88% en utilisant PROXI$_I$ et les comparaisons ont révélé qu'ils étaient significativement plus précis en utilisant PROXI$_I$ que toute autre technique ($p < .001$) excepté PROXI$_V$. Les participants ont été précis à environ 80% en utilisant les variantes de StressViz et aucune différence significative n'a été trouvée entre ces variantes. Les participants ont été précis à 84% en utilisant PROJVIZ, ce qui est significativement plus précis (à environ 4%) qu'en utilisant les variantes de StressViz ($p < .05$). De la même façon, les participants ont été significativement plus précis en utilisant les variantes de ProxiViz que les variantes de StressViz ($p < .005$). En revanche, aucune différence significative n'a été trouvée entre PROJVIZ et les variantes de ProxiViz excepté PROXI$_I$. Et entre les variantes de ProxiViz, seulement PROXI$_P$ est significativement moins précise que PROXI$_I$ ($p < .05$). Les participants ont été précis à 87% sur des projections composées principalement de déchirures, ce qui est significativement plus précis de 8% que sur des projections présentant des faux voisinages ($p < 0.0001$). Les participants ont été précis à 92% sur des jeux de données synthétiques, ce qui est 18% significativement plus précis que sur des jeux de données réelles ($p < 0.0001$).

Temps de réponse

En termes de temps de réponse, l'ANOVA révèle un effet significatif du facteur Technique ($F_{6,120} = 19.3, p < .0001$). Les comparaisons Post-hoc ont montré que les participants étaient significativement plus rapides avec PROJVIZ (15s) qu'avec les autres techniques : STRESS$_P$ (21s), STRESS$_I$ (23s), STRESS$_V$ (25s), PROXI$_P$ (26s), PROXI$_I$ (29s) and PROXI$_V$ (33s). PROXI$_V$ était significativement plus lent que toutes les autres Techniques ($p < 0.001$), excepté PROXI$_I$ qui était également significativement plus lent que STRESS$_I$, STRESS$_P$ and PROJVIZ ($p < 0.001$). PROXI$_P$ était significativement plus lent que STRESS$_P$ et PROJVIZ ($p < 0.05$). Aucun effet significatif du type d'artefacts n'a été trouvé en termes de temps de réponse.

Préférences utilisateurs

Chaque participant a rempli un questionnaire après l'évaluation pour classer les techniques et donner leurs préférences sur chacune en les triant selon une échelle de Likert de 1 (pas du tout aimé) à 7 (beaucoup aimé), en passant par 4 (sans opinion) en fonction de différents critères. Dans l'ensemble les participants on préféré PROXI$_I$ (11 participants). PROXI$_V$ était la seconde technique préférée (7 participants) et la suivante était PROXI$_P$ (3 participants). Les principales raisons justifiant ces choix étaient : l'esthétique, la facilité d'interprétation et l'interactivité. ProxiViz était jugé plus facile à interpréter et plus rassurante au niveau des réponses que StressViz. PROXI$_V$ a été également préférée car les cellules de Voronoï permettent de mieux distinguer les frontières entre les zones de couleurs différentes. Les participants se sont sentis significativement plus rapide et plus précis avec PROXI$_I$, PROXI$_V$ et PROJVIZ que les autres techniques (médiane ≥ 6 et $p < .05$). PROXI$_I$ était jugée globalement plus esthétique que les autres techniques et plus simple à utiliser (médiane $= 7$ et $p < .0001$), devant PROXI$_V$ (médiane $= 6$ et $p < .005$). Les participants ont trié les techniques par rapport à 3 critères : points correctement séparés, modérément séparés, ou non séparés (bruit). Dans chaque cas, PROXI$_I$ et PROXI$_V$ ont obtenus les meilleurs résultats (médiane de rang ≥ 2). Ces résultats qualitatifs sont cohérents avec les résultats quantitatifs, en particulier PROXI$_I$ est également la meilleure technique du point de vue des préférences.

4.5 Discussion

Les résultats montrent que la technique ProxiViz utilisant la couleur interpolée est significativement plus précise que les autres techniques. C'est également la technique que les participants ont le plus préféré pour ses caractéristiques esthétiques et sa facilité d'interprétation. Comparé à ProjViz, StressViz est moins précis et plus long à utiliser. Dans l'ensemble, les hypothèses sont vérifiées par nos résultats. Dans cette section, on discute les résultats par rapport aux questions de recherche initiales.

Q1. ProxiViz et StressViz sont-ils plus précis que ProjViz pour une tâche de clustering visuel ?

PROXI$_I$ est la seule technique qui soit significativement plus précise que PROJVIZ. Tous les résultats sont confirmés par les résultats qualitatifs : les participants se sont sentis plus rapides et plus précis avec PROXI$_I$ et cette technique a été jugée plus esthétique et simple à utiliser. Donc PROXI$_I$ est la meilleure technique pour un clustering visuel efficace à la fois d'un point de vue quantitatif et qualitatif. Néanmoins, nous observons qu'il y a environ 4% de différence entre PROXI$_I$ et PROJVIZ, ce qui est meilleure qu'entre les autres techniques, toutefois c'est un gain de précision qui reste faible. **H1** est partiellement validée car seulement PROXI$_I$ est significativement plus précis que PROJVIZ et les variantes de StressViz. Par rapport aux résultats de StressViz, qui ne sont pas significativement différents de ceux de ProjViz, on peut supposer que l'information de stress n'est pas suffisamment précise pour obtenir de meilleurs résultats que PROJVIZ. Les participants ont peut être fait des erreurs d'interprétation à cause de la difficulté à comprendre l'information de stress.

Q2. L'encodage couleur impacte-t-il la précision de StressViz et ProxiViz ? C'est à dire un encodage couleur est-il meilleur qu'un autre ?

Par rapport à ProxiViz, **H2a** est partiellement validée car une seule différence significative a été trouvée en termes de précision entre PROXI$_I$ et PROXI$_P$, et une seule différence significative a été trouvée en termes de temps de réponse entre PROXI$_V$ et PROXI$_P$. Même si aucune différence significative n'a été trouvée en termes de précision et de temps de réponse entre les encodages de couleur de PROXI$_I$ et PROXI$_V$, les participants ont été significativement plus précis et rapide avec PROXI$_I$ plutôt que PROXI$_V$.

On peut expliquer les faibles performances de PROXI$_V$ en termes de temps de réponse par le fait que la taille des cellules de Voronoi ne représente pas concrètement d'information, mais est juste lié à la disposition des points, ce qui peut créer un biais ayant perturbé les perceptions des participants. Néanmoins, certains participants se sont sentis plus précis avec PROXI$_V$ car cela permet de mieux distinguer les frontières entre les clusters que l'interpolation de la couleur. Ceci est soutenu par l'absence de différence significative en termes de précision entre PROXI$_V$ et PROXI$_I$.

Par rapport à StressViz, **H2b** est partiellement validée car aucune différence significative n'a été trouvé en termes de précision et de temps de réponse entre STRESS$_P$, STRESS$_V$ et STRESS$_I$. Aucun encodage couleur représentant l'information de stress n'a été plus performant qu'un autre. On peut supposer que l'information de stress est trop difficile à utiliser car elle donne des indications sur le bon ou mauvais positionnement des points mais n'indique pas où ils devraient être

positionnés. Nous avons également obtenu des retours sur l'échelle de couleur de Tominski et al. [227] utilisée pour ProxiViz qui a été jugée inconfortable par les participants du fait de leur manque d'habitude par rapport à l'ordre des couleurs.

Q3. L'interactivité de ProxiViz impacte-t-elle fortement les temps de réponse par rapport à ProjViz ?

H3 est partiellement vérifiée. En effet, les participants ont été significativement plus rapide avec PROJVIZ qu'avec les autres techniques (1.5× plus rapide en moyenne que StressViz et 1.9× plus rapide en moyenne que ProxiViz). Des différences significatives ont été trouvées en termes de temps de réponse entre PROXI_P, PROXI_V, PROXI_I et STRESS_P, STRESS_V et STRESS_I. Les participants ont été significativement plus lents avec PROXI_V plutôt que STRESS_P, STRESS_V et STRESS_I. De même avec PROXI_I plutôt que STRESS_I et STRESS_P. Et seulement significativement plus lents avec PROXI_P plutôt que STRESS_P. On peut supposer que les différences entre ProxiViz et StressViz viennent de l'interactivité nécessitée par ProxiViz. Mais les faibles écarts de temps de réponses tendent à montrer que le processus d'interprétation de l'information de stress est complexe et que les participants ont eu du mal à prendre une décision rapidement.

Q4. Le type d'artefacts topologiques présents impacte-t-il la précision du clustering visuel ?

H4 est vérifiée. On observe des différences significatives en termes de précision entre les deux types d'artefacts. Cependant aucune différence significative n'a été trouvée en termes de temps de réponse. Les participants ont été plus précis sur les projections avec des déchirures que sur celles avec des faux voisinages. Pour permettre d'effectuer un clustering visuel plus précis, il faut donc éviter des algorithmes de projection qui introduisent des faux voisinages.

Ceci peut s'expliquer par le faite que durant l'exploration pour trouver des clusters, il faut d'abord utiliser l'information du voisinage local tel qu'il est perçu en 2D. Or ce voisinage est plus fidèle au voisinage d'origine avec des déchirures qu'avec des faux voisinages. L'information de proximité 2D peut être corrigée avec des retours sur les proximités d'origine dans l'espace de données mais il est plus aisé de voir que deux zones 2D distantes sont en fait plus proches dans l'espace des données, que de clarifier une zone de faux voisinages où chaque point doit être pris en compte un à un.

4.6 Conclusion

Dans ce chapitre, nous avons décrit une expérience contrôlée visant à quantifier l'efficacité de l'ajout d'informations relatives aux artefacts afin d'aider le clustering visuel. Les résultats montrent que ProxiViz avec la coloration interpolée est significativement plus précis que les autres variantes ainsi que les autres techniques pour une tâche d'énumération de clusters. On remarque également que les projections avec des déchirures obtiennent globalement des résultats significativement meilleurs que celles avec des faux voisinages. Cependant la portée de ces résultats se limite à une tâche de clustering visuel et à des algorithmes de projection favorisant un seul type précis d'artefacts de projection. Le chapitre suivant présente une seconde expérience contrôlée visant à confirmer l'efficacité de ProxiViz sur un panel plus large de tâches d'analyse visuelle et ainsi que sur n'importe quelle projection, indépendamment de sa qualité en termes d'artefacts.

5

Evaluation de ProxiViz sur différentes tâches d'analyse visuelle

L'expérience précédente montre que ProxiViz avec une coloration interpolée des proximités d'origine est la technique la plus précise et la plus appréciée pour réaliser une tâche d'analyse visuelle sur des projections présentant soit des déchirures, soit des faux voisinages. La portée des résultats de cette expérience étant limitée, nous avons réalisé une deuxième expérience contrôlée visant à confirmer l'efficacité de ProxiViz sur un panel plus large de tâches d'analyse visuelle ainsi que sur n'importe quelle projection, indépendamment de sa qualité. Ce chapitre présente une expérience contrôlée comparant ProxiViz, utilisant une coloration interpolée des proximités d'origine, avec la projection, sans ajout d'information. Différents niveaux de difficulté sont pris en compte en fonction de la présence ou non d'artefacts par rapport à la tâche d'analyse visuelle considérée, à savoir la validation d'outliers de données ou d'outliers de classes, la vérification du clustering ou l'énumération des clusters.

5.1 Motivation

L'expérimentation contrôlée précédente a permis d'obtenir des retours quantitatifs et qualitatifs sur l'encodage visuel de ProxiViz. Il en ressort que l'utilisation d'une coloration interpolée des proximités d'origine est l'encodage le plus précis et le plus apprécié pour une tâche de clustering visuel. Cette évaluation a également mis en évidence que les algorithmes de projection favorisant des déchirures, plutôt que des faux voisinages, permettaient d'améliorer la précision du clustering visuel. Cependant le contrôle du type d'artefacts restreint la portée des résultats à des algorithmes favorisant un type précis d'artefact. De plus, ces artefacts de projection peuvent impacter différemment chaque tâche d'analyse visuelle, en particulier des tâches d'analyse locales comme la validation d'outliers.

Nous avons donc réalisé une seconde expérimentation contrôlée afin d'étudier l'impact des artefacts sur l'analyse visuelle des projections et d'évaluer si ProxiViz permet d'être significativement plus précis sur n'importe quelle projection, indépendamment des artefacts de projection. Dans cette expérience, nous comparons *ProxiViz*, utilisant une coloration interpolée des proximités d'origine, avec la projection sans ajout d'information *ProjViz*. Nous ne considérons pas l'ajout d'informations de stress car les résultats de l'expérience précédente ont montré que l'affichage du stress ne permettait pas d'aider au clustering visuel de la projection.

Dans ce chapitre, nous décrivons l'expérience contrôlée que nous avons réalisé afin de répondre aux questions suivantes :

- Q.1 ProxiViz permet-il d'aider l'analyse visuelle des projections indépendamment de la qualité de celles-ci ?
- Q.2 Comment les artefacts de projection impactent-ils les tâches d'analyse visuelle des projections ?
- Q.3 Est ce que l'interactivité de ProxiViz est rentable par rapport au gain potentiel qu'elle apporte en précision ?

5.2 Expérience contrôlée

Dans cette section, nous décrivons d'abord les différentes tâches d'analyse visuelle considérées dans l'expérience, puis sa conception générale et son déroulement, avant d'aborder les résultats.

5.2.1 Tâches d'analyse visuelle

A partir de la taxonomie des tâches d'analyse visuelle (cf. section 2.4), nous avons considéré deux contextes : exploratoire et confirmatoire. Certaines tâches, comme l'extraction de clusters ou la détection d'outliers, nécessitent d'explorer l'ensemble de la projection. Il est difficile d'utiliser de telles tâches exploratoires dans le cadre d'une expérience contrôlée, car il est difficile de maîtriser le temps de réponse des participants. De plus, il est difficile ensuite de juger de la qualité des réponses par rapport à la réalité dans les données.

Comme la validation des réponses à ces tâches exploratoires introduirait de nouveaux biais, nous nous concentrons sur des tâches de validation impliquant des outliers et clusters dans les contextes exploratoires et confirmatoire. Nous considérons également la tâche d'énumération de cluster comme dans l'expérience précédente, pour confirmer les résultats de ProxiViz sur d'autres jeux de données plus complexes.

Nous considérons les tâches de validation suivantes (Figure 1.4) :

Vérification d'outliers de données : Pour un point mis en valeur sur la projection, les participants doivent choisir une réponse oui/non à la question : "est-ce un outlier de données ?", c'est-à-dire ce point est-il isolé dans l'espace des données ?

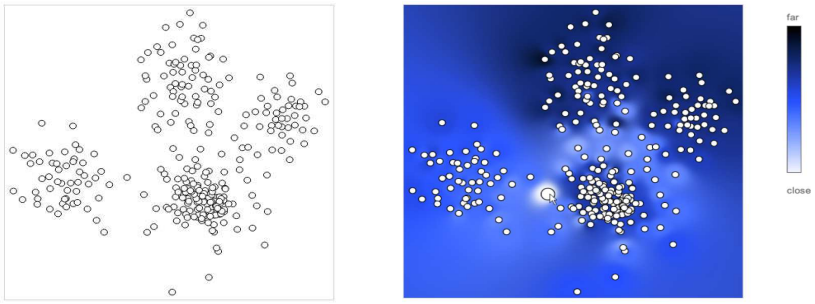
Vérification du clustering : Pour deux ensembles de points colorés différemment et mis en valeur sur la projection, les participants doivent choisir une réponse oui/non à la question : "Est ce que ces deux jeux de points appartiennent au même cluster ?".

Enumération des clusters : Les participants doivent choisir une réponse à la question : "Combien de clusters comptez vous ?" en choisissant une réponse entre 1 et 7, afin d'énumérer les clusters distinguable visuellement sur la projection.

Vérification d'outlier de classe : Pour un point mis en valeur sur la projection colorée en fonction des étiquettes de classes, les participants doivent choisir une réponse oui/non à la question : "Est ce un outlier de classe ?", c'est-à-dire ce point appartient-il à un cluster de points d'une autre classe ?

5.2.2 Techniques

Nous considérons les techniques ProjViz et ProxiViz telles que décrites dans l'expérience précédente. Pour ProxiViz, les retours utilisateurs, obtenus sur l'échelle de couleur de Tominski et al. [227], nous amènent à utiliser une autre échelle utilisant une seule teinte de couleur. Nous considérons ainsi un dégradé uniforme de bleu qui varie en brillance. On considère dans cette expérience seulement ProxiViz utilisant une interpolation de la coloration (Figure 5.1).



(a) ProjViz (b) ProxiViz

FIGURE 5.1 – Résumé des techniques utilisées dans l'expérience.

5.2.3 Jeux de données

Nous avons utilisé les 3 mêmes jeux de données réelles que dans l'expérience précédente. Excepté pour le jeu de données *letters*, où nous avons utilisé de nouveaux échantillons composés de 6 lettres (A, E, I, O, U, Y) avec 30 individus choisis aléatoirement pour chaque lettre, afin de complexifié son clustering visuel. Dans cette expérience, nous considérons également 3 jeux de données réelles supplémentaires :

Pen-Based Recognition of Handwritten Digits (*pen-digits* de [87]) : 34 personnes ont écrit 250 chiffres (0-9) dans un ordre aléatoire à l'intérieur de rectangles sur une tablette de 500 par 500 pixels. Chaque individu est résumé par un ensemble de 16 dimensions. Nous avons utilisé un échantillon composé de 5 chiffres (0, 1, 3, 4, 5) avec 50 individus choisis aléatoirement pour chaque chiffre.

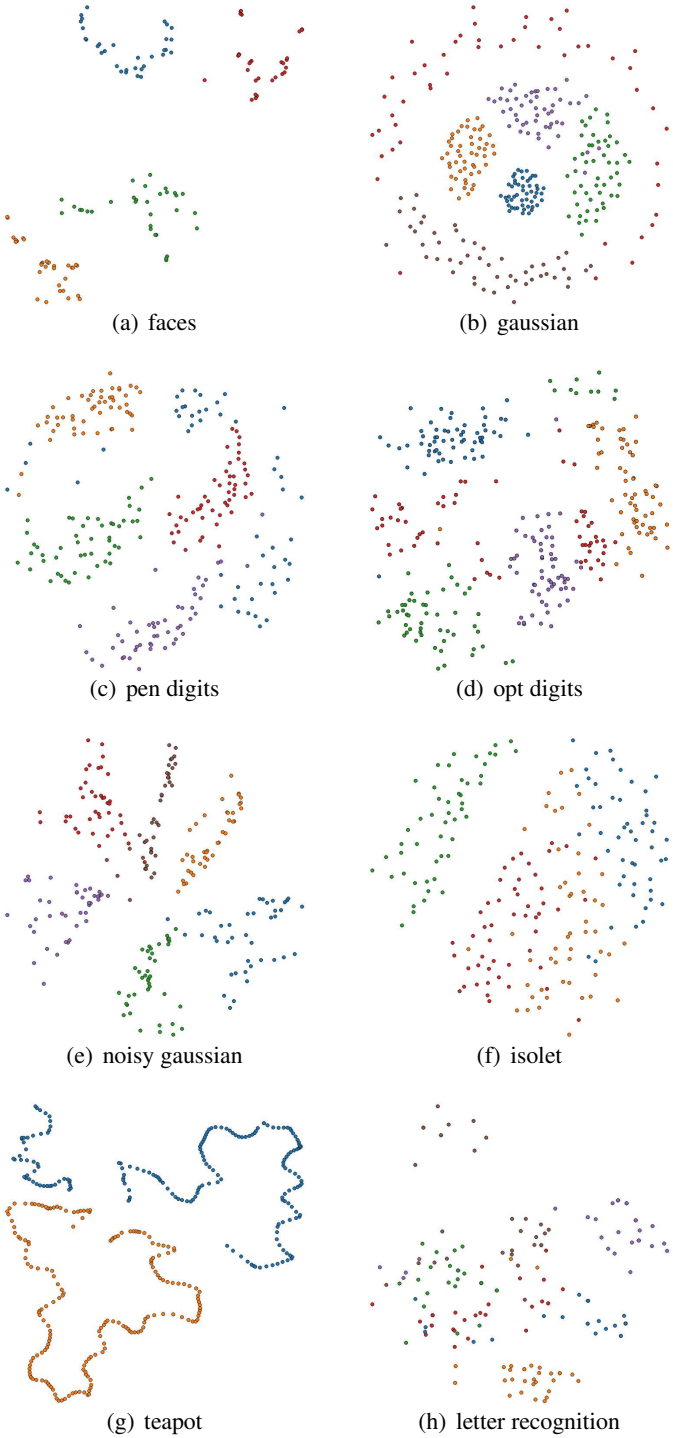
Optical Recognition of Handwritten Digits (*opt-digits* de [87]) : 32x32 bitmaps normalisés de chiffres écrits à la main (0-9) ont été extraits à partir de chiffres écrits à la main par 43 personnes. Les bitmaps ont été ensuite divisés en blocs non chevauchant pour générer une matrice 8x8 où chaque élément est un entier dans un intervalle de 0..16. Nous avons utilisé un échantillon composé de 5 chiffres (1, 5, 7, 8, 9) avec 50 individus choisis aléatoirement pour chaque chiffre.

ISOLET - Isolated Letter Speech Recognition (*isolet* de [87]) : 150 sujets ont épelé chaque lettre de l'alphabet deux fois et ont été groupés en 5 ensembles de 30 sujets. Chaque individu est résumé par un ensemble de 617 dimensions continues comme le coefficient spectral, le contour des formes, etc.. Nous avons utilisé un échantillon composé de 4 lettres (A, B, C, D) avec 50 individus choisis aléatoirement pour chaque lettre.

Nous avons également généré 2 jeux de données synthétiques avec la même méthode que dans l'expérience précédente. Cependant nous ajoutons un bruit uniforme aléatoire sur les 10 dernières dimensions du second jeu de données, afin de le complexifier. Nous avons ensuite projeté tous ces jeux de données en utilisant 2 algorithmes différents : classical MDS et NeRV (avec λ=0). Nous avons vérifié visuellement la séparation des clusters et sélectionné les projections dans le but d'équilibrer la difficulté des questions. Pour résumer, nous considérons 2 jeux de données synthétiques et 6 jeux de données réels dans l'expérience (Figure 5.2 et Figure 5.3).

Nom	Dimensions	Instances	Clusters	MDS
synthétique1	100	300	6	NeRV
synthétique2	100	240	6	NeRV
pen-digits	16	250	5	classical
opt-digits	64	300	6	NeRV
isolet	617	200	4	classical
letters	16	180	6	classical
faces	960	128	4	NeRV
teapot	192	365	2	NeRV

FIGURE 5.2 – Résumé des jeux de données utilisés dans l'expérience.



(a) faces (b) gaussian

(c) pen digits (d) opt digits

(e) noisy gaussian (f) isolet

(g) teapot (h) letter recognition

FIGURE 5.3 – Projections des jeux de données utilisés dans l'expérience.

5.2.4 Niveau de difficulté

Nous avons exploré visuellement chaque projection en utilisant ProxiViz afin de détecter les artefacts. Ensuite nous avons défini les questions/réponses manuellement pour chaque tâche et jeu de données dans le but d'équilibrer les questions en deux catégories : facile et difficile. Les questions faciles sont des questions auxquelles ont peut répondre sans ajout d'information et les questions difficiles mettent en jeu des artefacts topologiques ou des problèmes de séparation des clusters. Nous avons utilisé des approches automatiques pour suggérer les possibles clusters et outliers, à l'aide de la librairie scikit-learn [171] :

Vérification d'outliers de données : Nous avons cherché des outliers de données pour chaque jeu de données. En utilisant les algorithmes One Class SVM [192] et Local Outlier Factor [35], nous avons déterminé deux ensembles d'outliers de données possibles. Ensuite nous avons sélectionné visuellement un outlier à l'intersection des deux ensembles pour différentes projections. Une question facile est une question mettant en jeu un outlier qui est clairement séparé du clusters de points 2D le plus proche. A l'inverse, une question difficile met en jeu à un outlier qui n'est pas séparé des points 2D, c'est-à-dire un faux voisinage.

Vérification du clustering : Nous avons sélectionné différents clusters et composantes de clusters pour chaque jeu de données. Ensuite nous avons sélectionné visuellement deux ensembles de points proches dans l'espace 2D et nous avons analysé leurs proximités dans l'espace des données. Une question difficile met en jeu deux ensembles de points correspondant à une déchirure si les points appartiennent au même cluster dans les données, ou bien correspondant à des faux voisinages si les points proviennent de deux clusters différents dans les données.

Enumération des clusters : Nous avons analysé visuellement la qualité du clustering 2D de chaque projection. Ensuite pour valider la vérité par rapport aux données, nous avons vérifié que chaque classe avait un cluster sous-jacent dans les données et que les clusters ne se chevauchaient pas trop entre eux. Une question difficile met en jeu des clusters 2D qui se chevauchent ou bien qui correspondent à un cluster séparé en différentes composantes non connexes.

Vérification d'outlier de classe : Nous avons extrait des outliers de classe pour chaque jeu de données. En utilisant les algorithmes KNN Classifier [59] et One Class SVM [192] sur chaque classe, nous avons déterminé visuellement un outlier de classe intéressant pour chaque projection. Une question difficile met en jeu un point qui est loin des autres points de sa classes en 2D, mais qui en réalité est une déchirure et non pas un outlier de classe.

5.2.5 Conception de l'expérience

Nous considérons une expérience contrôlée avec des mesures répétées (repeated-measure) avec les facteurs intra-sujets (within-subjects) suivants : Technique et Difficulté. L'ordre des blocs et des tests est contre-balancé en utilisant un carré latin, de manière à éviter les effets d'apprentissage. Nous avons également changé l'orientation et la symétrie des projections entre les blocs. Chaque participant a été confronté successivement à 2 blocs (un pour chaque technique) de 24 trials (3 tâches $\times$ 8 projections) et ensuite 2 blocs composés de 8 tests pour la tâche liée à la vérification de l'outlier de classe. Nous avons ordonné les blocs avec ProjViz en premier, car comme ProxiViz

montre plus d'information, les participants pourraient être biaisés par un effet de mémorisation en utilisant ProjViz ensuite. En résumé, l'expérience comprend :

4	projections	×
2	difficultés	×
3	tâches	×
2	techniques (blocs)	+
4	projections	×
2	difficultés	×
1	tâche	×
2	techniques (blocs)	=
64	tests par participants	×
24	participants	=
1,536	**tests au total**	

5.2.6 Participants et procédure

24 participants (8 femmes et 16 hommes) de 23 à 37 ans (28 en moyenne) de deux laboratoires spécialisés en analyse de données ont complété l'évaluation. Ils avaient tous une formation en informatique mais dans différents domaines : statistiques, analyse du signal, analyse d'image, apprentissage, data mining. Ils avaient différents niveaux d'expertise par rapport à l'analyse de données (12 étaient doctorants). Ils étaient rarement confrontés à des projections (15 y étaient confrontés au moins une fois par an, 5 une fois par mois, 4 jamais) et disposaient de peu de connaissances sur les projections (les techniques de réduction de dimension connues étaient l'ACP, kernelPCA, ISOMAP, Kohonen SOM). Ils étaient habitués à utiliser les visualisations statiques proposés par différents systèmes d'analyse exploratoire (matlab, R, excel, gnuplot, matplotlib). Ils ont tous déclaré avoir une vision normal sans problèmes de perception des couleurs.

Nous avons enregistré les réponses ainsi que les temps de réponse de chaque participant pour chaque question. Les participants ont répondu aux questions en sélectionnant une réponse dans un questionnaire à choix multiples oui/non ou avec un intervalle de 1 à 7 clusters et en cliquant ensuite sur un bouton de validation à l'aide de la souris. Le temps a été enregistré à partir du moment où la projection apparaissait à l'écran et l'enregistrement se stoppait lorsque le participant avait validé une réponse. Après chaque validation, nous avons également recueilli le niveau de confiance par rapport à la réponse (entre 5 niveaux : pas sûr, septique, ni sûr ni septique, confiant, très confiant). Entre chaque bloc, les participants pouvaient faire une pause. Nous avons imposé une durée de 30-60 secondes maximum pour chaque test, pour une durée totale de l'évaluation d'environ 30 minutes, précédé par 20 minutes de présentation de l'évaluation et un entrainement. Le système d'évaluation à été implémenté en `d3.js` [32] et affiché en plein écran sur un moniteur de résolution 1440 × 900 d'un MacBook Pro.

Après une courte introduction sur les artefacts et l'analyse visuelle, nous avons vérifié que les participants avaient compris comment utiliser les techniques et appréhendé les définitions d'outlier de données, de cluster et d'outlier de classe. Nous avons également clarifié le fait que le clustering dépend du modèle des données. En effet, les clusters peuvent être vues comme des "blobs" avec

des densités différentes ou comme des sous-ensembles d'une géométrie complexe (une variété) sur laquelle reposent différentes distributions statistiques. Nous avons entrainé les participants à utiliser ProxiViz sur un jeu de données simple, généré à partir de Gaussiennes, pour chaque tâche. Nous les avons laissé expliquer leur raisonnement pour chaque réponse de manière à valider leur compréhension du système. Si nécessaire, nous leur avons expliqué les problèmes de leur raisonnement et montré comment obtenir un raisonnement correct.

Nous avons utilisé 6 autres participants comme pilotes pour mettre à jour la procédure de l'expérience et vérifier la cohérence des résultats par rapport à nos attentes. Avant l'évaluation, les participants ont rempli un questionnaire pour obtenir des informations démographiques. Après l'expérience, les participants ont également rempli un questionnaire pour recueillir leurs préférences sur les différentes techniques et leur ressenti sur l'expérience.

5.2.7 Hypothèses

Nous supposons que es participants seront significativement plus précis avec ProxiViz plutôt que ProjViz, comme dans l'expérience précédente. Et surtout, nous supposons que cette technique obtiendra des résultats similaires sur les questions faciles et difficiles, pour chaque tâche, là où ProjViz est supposée être significativement moins précise sur des questions difficiles. Nous supposons également que les participants seront significativement plus lents avec ProxiViz, mais sans différence entre les questions faciles et difficiles. Basé sur les questions de recherche introduites précédemment, nos hypothèses sont les suivantes :

H1 Pour chaque tâche, nous supposons que ProxiViz est plus précis que ProjViz pour chaque type de projection.

H2 Pour chaque tâche, ProxiViz est plus précis que ProjViz pour des questions difficiles et aussi efficaces que ProjViz dans des questions faciles.

H3 Pour chaque tâche, les participants seront significativement plus rapides en utilisant ProjViz.

5.3 Résultats

Cette section décrit les résultats statistiques en termes de précision et de temps de réponse (Figure 5.4). Nous avons transformé l'erreur en un pourcentage de précision pour les tâches qui supposent une réponse oui/non. La précision est soit 100% soit 0%, la réponse est ainsi soit correcte soit fausse. Pour la tâche d'énumération de clusters, nous avons utilisé la même formule de précision que dans l'expérience précédente.

Comme nous considérons des facteurs intra-sujets (avec répétition des mesures) et que pour toutes les tâches la précision ne suit pas une distribution de loi Normale, alors nous utilisons le test non-paramétrique de Friedman pour analyser la variance ainsi que des tests pairés de Wilcoxon (de rang signé) pour analyser 2 distributions indépendantes non-paramétriques. Le temps de réponse est log-transformé pour mieux correspondre à une distribution de loi Normale. On utilise ensuite l'ANOVA pour analyser la variance et des t-tests pairés pour comparer les temps de réponse. Les résultats de confiance sont comparés avec des tests pairés de Wilcoxon (de rang signé).

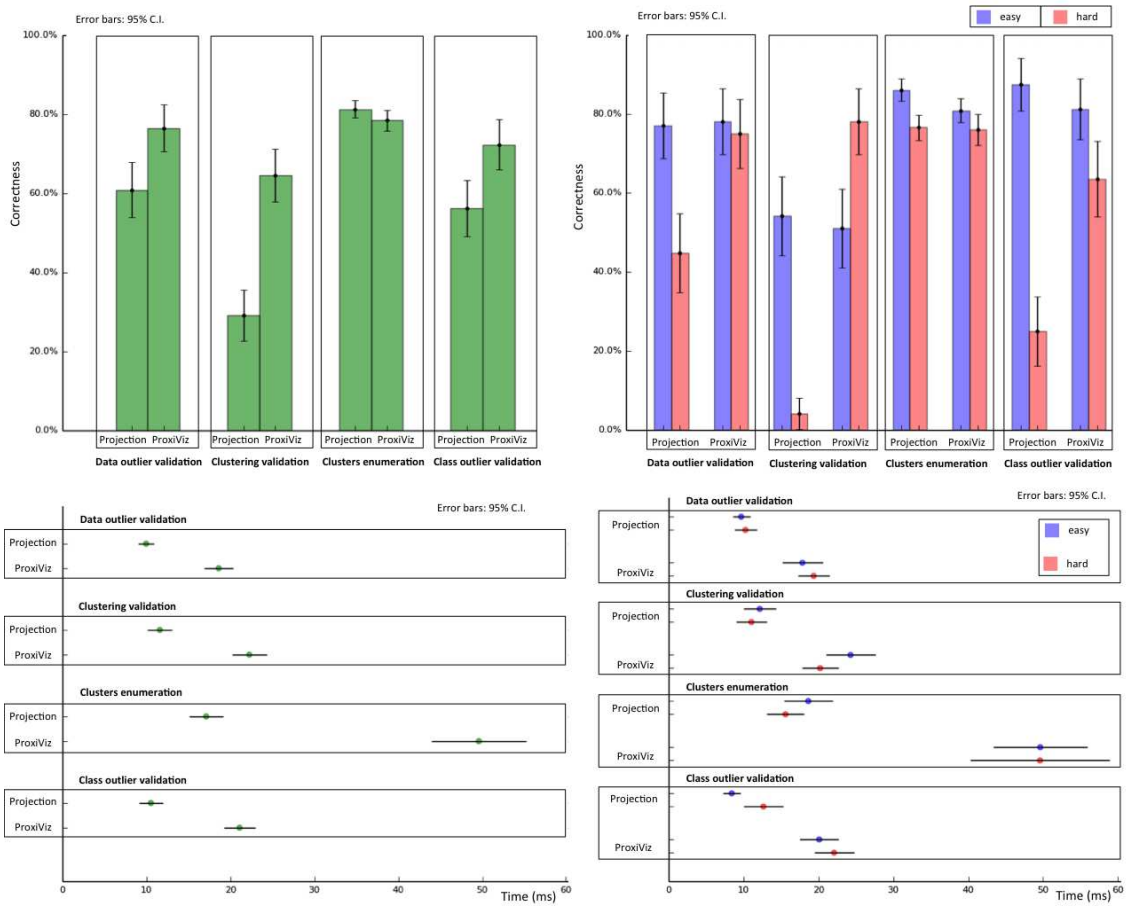


FIGURE 5.4 – Résultats en termes de moyennes et écarts types de précision (en %) et de temps de réponse (en secondes) pour chaque technique.

Précision - Vérification d'outliers de données : En termes de précision, le test de Friedman révèle un effet significatif du facteur Technique ($\chi^2(1) = 10, p < .001$) et des facteurs Technique et Difficulté ensembles ($\chi^2(3) = 32.77, p < .0001$). Les participants ont été précis à 77% en utilisant ProxiViz, ce qui est 16% significativement plus précis qu'avec ProjViz ($p < .005$). Aucune différence significative n'a été trouvé entre les deux techniques pour les questions faciles et les participants ont été précis à 77% avec chacune des techniques. Cependant pour des questions difficiles, les participants ont été précis à 75% en utilisant ProxiViz, ce qui est 30% significativement plus précis qu'avec ProjViz ($p < .0001$). Pour ProxiViz, il n'y a pas de différences significatives entre les questions faciles et difficiles, mais les participants ont été 33% significativement plus précis sur des questions faciles que des questions difficiles avec ProjViz ($p < .0001$).

Précision - Vérification du clustering : En termes de précision, le test de Friedman révèle un effet significatif du facteur Technique ($\chi^2(1) = 42, p < .0001$) et des facteurs Technique et Difficulté ensembles ($\chi^2(3) = 105, p < .0001$). Les participants ont été précis à 65% avec ProxiViz, ce qui est 35% significativement plus précis qu'avec ProjViz ($p < .0001$). Aucune différence significative n'a été trouvé entre les deux techniques pour des questions faciles et les participants ont été précis à 54% avec ProjViz et à 51% avec ProxiViz. Cependant pour des questions difficiles, les participants ont été précis à 78% avec ProxiViz et à 4% pour ProjViz, ce qui est significativement différent ($p < .0001$). Pour ProxiViz, les participants ont été 27% significativement plus précis sur des questions difficiles que sur des questions faciles ($p < .0001$). Pour ProjViz, les participants ont été 50% significativement plus précis sur des questions faciles que sur les difficiles ($p < .0001$).

Précision - Enumération des clusters : En termes de précision, le test de Friedman ne révèle pas un effet significatif du facteur Technique ($\chi^2(1) = 2, p < .1$), mais en revanche il révèle un effet des facteurs Technique et Difficulté ensembles ($\chi^2(3) = 105, p < .0001$). Les participants ont été précis à 78% avec ProxiViz et à 81% avec ProjViz, ce qui n'est pas significativement différent ($p > .05$). Pour des questions faciles, les participants ont été précis à 86% avec ProjViz, ce qui est 6% significativement plus précis qu'avec ProxiViz ($p < .01$). Cependant aucune différence significative n'a été trouvé entre les techniques pour des questions difficiles et les participants ont été précis à 76% avec chacune des techniques. Pour ProxiViz, il n'y a pas de différence significative entre les questions faciles et difficiles, mais les participants ont été 10% significativement plus précis sur les questions faciles que sur les difficiles avec ProjViz ($p < .0001$).

Précision - Vérification d'outlier de classe : En termes de précision, le test de Friedman révèle un effet significatif du facteur Technique ($\chi^2(1) = 13, p < .0001$) et des facteurs Technique et Difficulté ensembles ($\chi^2(3) = 98, p < .0001$). Les participants ont été précis à 72% avec ProxiViz, ce qui est 16% significativement plus précis qu'avec ProjViz ($p < .0005$). Pour des questions faciles, les participants ont été précis à 87% avec ProjViz, ce qui est 6% significativement plus précis qu'avec ProxiViz ($p < .01$). Sur les questions difficiles, les participants ont été précis à 63% avec ProxiViz, ce qui est 38% significativement plus précis qu'avec ProjViz ($p < .0001$). Pour ProxiViz, les participants ont été 18% significativement plus précis sur les questions faciles que sur les difficiles ($p < .0001$). Pour ProjViz, les participants ont été 62% significativement plus précis sur les questions faciles que sur les difficiles ($p < .0001$).

Temps de réponse - Vérification d'outliers de données : En termes de temps de réponse, l'ANOVA révèle un effet significatif du facteur Technique ($F_{1,382} = 75.3, p < .0001$) mais aucune interaction significative entre les facteurs Technique et Difficulté. Les participants ont été significativement 2 $\times$ plus lents avec Proxiviz qu'avec ProjViz ($p < .0001$) qui leur à nécessité en moyenne environ 10 secondes par projection. Les participants ont été significativement 1.8 $\times$ plus lents avec Proxiviz qu'avec ProjViz ($p < .0001$) à la fois pour les questions faciles et les difficiles nécessitant en moyenne environ 10 secondes par projection. Aucune différence significative n'a été trouvé en temps de réponse entre les questions faciles et difficiles pour chaque technique.

Temps de réponse - Vérification du clustering : En termes de temps de réponse, l'ANOVA révèle un effet significatif du facteur Technique ($F_{1,382} = 67.8, p < .0001$) mais aucune interaction significative entre les facteurs Technique et Difficulté. Les participants ont été significativement 1.8 $\times$ plus lents avec Proxiviz qu'avec ProjViz ($p < .0001$) qui leur à nécessité en moyenne environ 12 secondes par projection. Les participants ont été significativement 2 $\times$ plus lents avec Proxiviz qu'avec ProjViz ($p < .0001$) à la fois pour les questions faciles et difficiles. En utilisant ProxiViz, les participants ont été significativement plus lents sur les questions faciles que sur les difficiles ($p < .05$) , mais aucune différence significative n'a été trouvé en temps de réponse entre les questions faciles et difficiles pour ProjViz.

Temps de réponse - Enumération des clusters : En termes de temps de réponse, l'ANOVA révèle un effet significatif du facteur Technique ($F_{1,382} = 112.6, p < .0001$) mais aucune interaction significative entre les facteurs Technique et Difficulté. Les participants ont été significativement 3 $\times$ plus lents avec Proxiviz qu'avec ProjViz ($p < .0001$) qui leur à nécessité en moyenne environ 17 secondes par projection. ProxiViz leur a nécessité environ 50 secondes par projection pour des questions faciles et difficiles, ce qui est significativement 2.6-3 $\times$ plus lent ($p < .0001$) qu'avec ProjViz. Aucune différence significative n'a été trouvée en temps de réponse entre les questions faciles et difficiles pour chaque technique.

Temps de réponse - Vérification d'outlier de classe : En termes de temps de réponse, l'ANOVA révèle un effet significatif du facteur Technique ($F_{1,382} = 76.7, p < .0001$) mais aucune interaction significative entre les facteurs Technique et Difficulté. Les participants ont été significativement 2 $\times$ plus lents avec Proxiviz qu'avec ProjViz ($p < .0001$) qui leur à nécessité en moyenne environ 11 secondes par projection. Les participants ont été significativement 1.8-2.5 $\times$ plus lents avec Proxiviz qu'avec ProjViz ($p < .0001$) à la fois pour les questions faciles et les difficiles. Aucune différence significative n'a été trouvé entre les questions faciles et difficiles pour chaque technique.

Confiance et préférences

Aucune différence significative n'a été trouvé en termes de confiance. Pour toutes les tâches, les participants ont été confiants dans leurs réponses excepté pour la tâche d'énumération des clusters pour laquelle ils ont été en moyenne ni confiant ni sceptique. Des corrélations ont été trouvées entre la confiance et la précision, ainsi qu'entre la confiance et le temps de réponse, mais elles sont trop faibles pour en tirer des conclusions (environ 0.1).

Globalement ProxiViz a été plus appréciée que ProjViz (20 participants). 18 participants se sont sentis plus confiants et ont préféré ProxiViz pour sa clarté. 18 participants se sont sentis plus rapides en utilisant ProjViz. 21 participants ont eu le sentiment de faire des réponses plus précises avec ProxiViz et 16 participants ont été satisfaits par leurs choix. Certain participants se sont sentis responsables des erreurs faites, car pour certaines projection ils avaient l'impression de ne pas être en mesure de décoder l'information additionnelle interactivement affichée.

5.4 Discussion

Dans cette section, nous discutons les résultats au regard des questions de recherche définies précédemment. Globalement, nos hypothèses sont vérifiées par les résultats.

Q.1 ProxiViz permet-il d'aider l'analyse visuelle des projections indépendamment de la qualité de celles-ci ?

Pour les tâches de vérification d'outliers de données et de classes, **H1** est vérifiée car ProxiViz est significativement plus précis à 16% que ProjViz. Pour la tâche de vérification du clustering également car ProxiViz est significativement plus précis à 33% que ProjViz. Ces résultats étaient attendus car ProxiViz met en valeur les relations locales de proximité. Cela montre que les participants ont été à même d'interpréter correctement la coloration. Cependant **H1** n'est pas vérifiée pour la tâche d'énumération des clusters, car on observe que ProxiViz n'est pas significativement plus précis que ProjViz. Ceci est également confirmé par une faible confiance des participants pour cette tâche.

Ce dernier résultat est surprenant car l'expérience précédente a montré que ProxiViz était significativement plus précis à 4% que ProjViz. Les participants ont été précis à 78% avec ProxiViz contre 88% dans l'expérience précédente et 81% contre 84% pour ProjViz. Ce résultat peut être dû au fait que l'expérience précédente contrôle le type d'artefacts et en particulier la considération de projections avec uniquement des déchirures peut avoir favorisé ProxiViz. Nous avons également utilisé dans cette seconde expérience un plus grand nombre de jeux de données réelles et les données synthétiques étaient plus complexes.

En particulier, nous avons remarqué durant l'évaluation que les participants se sont sentis perdus sur cette tâche en utilisant ProxiViz, car d'un jeu de données à l'autre les distributions des distances n'étaient pas semblables et il fallait faire un effort pour adapter l'interprétation des colorations à chaque test. Certains participants ont également trouvé difficile d'explorer la projection avec ProxiViz, car ils avaient du mal à interpréter les changements de couleur d'une référence à une autre. De plus, il était parfois difficile de juger visuellement comment délimiter les clusters en fonction des variations de brillance de la coloration. Nous avons aussi remarqué que certains participants ont utilisé les positions 2D plutôt que la coloration de ProxiViz, lorsque celle-ci était difficile à interpréter. Nous avons également noté des problèmes de mémorisation des clusters révélés avec ProxiViz. En effet, certains participants recomptaient plusieurs fois les clusters, ce qui suggère le besoin d'outils pour aider à conserver des traces de leurs découvertes.

Globalement, les résultats montre que les participants ont été relativement précis dans leurs analyses des colorations de ProxiViz (entre 60% et 80% de précision). ProxiViz est également significativement plus fiable que la projection sans ajout d'information pour des tâches impliquant l'analyse de proximités locales comme la vérification d'outliers ou de clusters. En revanche pour des tâches plus globales comme l'énumération des clusters, ProxiViz n'apporte pas significativement de gain en précision par rapport à la projection qui fournit déjà un aperçu global du clustering relativement précis (environ 80% de précision sur les projections des jeux de données considérés).

Q.2 Comment les artefacts de projection impactent-ils les tâches d'analyse visuelle des projections ?

Pour les tâches de vérification d'outliers de données et de classes, **H2** est vérifiée car ProxiViz est significativement plus précis à 30-38% que ProjViz sur les questions difficiles. Pour la tâche de vérification du clustering également, car ProxiViz est significativement plus précis à 74% que ProjViz sur des questions difficiles. Néanmoins pour la tâche de vérification d'outliers de classe, on observe que les 6% de différence de précision sur les questions faciles entre ProxiViz et ProjViz sont significatifs.

Pour la tâche d'énumération des clusters, **H2** est partiellement vérifiée car ProxiViz est aussi précise que ProjViz sur les questions difficiles. Néanmoins on observe que les 6% de différence de précision sur les questions faciles entre ProxiViz et ProjViz sont significatifs. Ce qui suggère que lorsque les clusters 2D sont biens séparés, la projection sans ajout d'information permet d'énumérer plus précisément les clusters qu'avec ProxiViz. Les proximités d'origine colorées interactivement sur la projection étant relatives à une référence, il faut décoder puis agréger mentalement cette information locale pendant le processus d'exploration afin d'obtenir un aperçu du clustering des données. La charge cognitive est donc plus importante, ce qui peut amener à faire plus d'erreurs d'analyse qu'avec la projection, pour laquelle les inférences à partir du clustering 2D sont directes mais pas toujours fiable à cause des artefacts.

Lorsque l'on considère toutes les tâches et techniques, on remarque que la précision maximum est proche de 90% et la précision minimum est autour de 5%. Ceci suggère que les questions n'étaient ni trop faciles ni trop difficiles et en particulier pour les tâches d'analyse locale impliquant une réponse "oui/non", l'analyse visuelle impliquant une prise de décision, même sur les questions faciles certains participants ont eu des conclusions différentes des autres. Un entraînement plus important des participants peut permettre de leur faire partager les mêmes critères visuels de décision concernant les outliers et les clusters, afin d'améliorer la précision des résultats.

Globalement, les résultats montrent que ProxiViz est plus robuste à l'influence des artefacts que la projection sans ajout d'information. Pour des tâches d'analyse locale, l'impact des artefacts sur la précision de l'analyse visuelle de la projection est important. Les analystes de données utilisant les projections sans ajout d'information doivent clairement être informés du risque important de parvenir à des conclusions erronés. Pour des tâches d'analyse globale du clustering visuel, l'impact des artefacts semble moins important. La projection statique est donc intéressante pour obtenir rapidement un aperçu approché du clustering des données. Toutefois de nouvelles techniques permettant "d'augmenter" les projections peuvent permettre d'améliorer la précision de l'analyse visuelle du clustering des données.

Q.3 Est ce que l'interactivité de ProxiViz est rentable par rapport au gain potentiel qu'elle apporte en précision ?

Pour les tâches de vérification des clusters et d'outliers de données et de classes, **H3** est vérifiée car ProxiViz est significativement 2 $\times$ plus lent que ProjViz sur les questions faciles et difficiles. Pour la tâche d'énumération des clusters, **H3** est également vérifiée, car ProxiViz est significativement 3 $\times$ plus lent que ProjViz sur les questions faciles et difficiles. Globalement aucune différence significative n'a été trouvé entre les questions faciles et difficiles pour chaque technique et tâche.

L'exploration avec ProxiViz pour des tâches d'analyse locales ne nécessite pas une grande différence de temps comparé à l'analyse de la projection sans ajout d'information. De plus, ce temps est indépendant des artefacts pour les deux techniques. ProxiViz est utile pour rentrer dans les détails des proximités et parvenir à des conclusions dans lesquelles on peut voir confiance et être satisfait, car elles résultent d'une exploration méthodique et pas seulement d'un aperçu qui peut manquer de précision. Ainsi de nouvelles techniques comme ProxiViz qui augmentent les projections doivent être développées en prenant les analystes en compte pour les conforter dans leurs tâches d'analyses et à les aider à mieux appréhender des données en grande dimension.

5.5 Conclusion

Dans ce chapitre, nous avons décrit une expérience contrôlée visant à quantifier l'efficacité de ProxiViz pour aider l'analyse de projections quelle que soit leur qualité en termes d'artefacts de projection. Cette expérience compare ProxiViz à la projection sans ajout d'information pour des tâches d'analyse locale et globale des proximités dans les données, impliquant à la fois des outliers et des clusters et pour différents niveaux de difficulté selon la présence ou non d'artefacts relativement à la tâche considérée. Les résultats montrent que ProxiViz est une technique interactive efficace pour des tâches d'analyse locale à partir d'une projection comme la vérification d'outliers ou de clusters. De plus, cette technique est robuste à l'influence des artefacts de projection alors que la projection sans ajout d'information ne l'est pas pour ces tâches. En revanche, pour des tâches d'analyse globale comme l'énumération des clusters, les résultats montrent que l'influence des artefacts semble moins importante et que ProxiViz n'apporte rien à la projection qui donne déjà un aperçu approché du clustering des données relativement satisfaisant. De plus, les participants ont rencontrés certaines difficultés dans leur analyse visuelle avec ProxiViz, comme détecter les variations de brillance dans la coloration de la projection ou mémoriser les structures sous-jacentes découvertes. Le chapitre suivant traite ces problématiques et propose une solution basée sur la métaphore de lentille [24] pour aider le travail d'analyse visuelle des projections.

6

ProxiLens
Exploration interactive des proximités d'origine

Les retours des participants lors des expériences décrites précédemment ont permis d'identifier différentes problématiques inhérentes à ProxiViz. La plupart de ces problèmes proviennent des artefacts de faux voisinages qui entâchent l'analyse du voisinage de la référence dans l'espace des données. Ce chapitre présente une exploration de l'espace de conception d'une lentille permettant de déformer localement la projection, afin de la nettoyer de ses faux voisinages et de mettre ainsi en valeur les proximités d'origine dans le voisinage d'un individu de référence. Nous introduisons des éléments de concept et discutons différents critères permettant de restreindre le champ des combinaisons possibles, en particulier au niveau de la représentation de cette lentille. Une implémentation de ce concept de lentille, que nous appellerons ProxiLens, est ensuite présentée et illustrée sur un jeu de données d'images pour les différentes tâches d'analyse visuelle.

6.1 Motivation

ProxiViz a pour objectif d'aider à l'analyse visuelle des projections et comme nous l'avons vu dans le chapitre précédent, cette technique permet de s'abstraire en partie de l'influence des artefacts topologiques pour des tâches d'analyse locale des proximités. Cependant cette technique n'est pas complétement indépendante des artefacts topologiques et en particulier les artefacts de faux voisinages posent des problèmes au niveau de la représentation et de l'interaction de navigation (cf. chapitre 3). De plus, nous avons remarqué des problèmes de mémorisation des structures sous-jacentes mises en évidence par ProxiViz.

En effet, nous avons considéré jusqu'ici la problématique *d'interpréter* correctement la projection, mais nous n'avons pas discuté les problèmes *d'extraction* des informations révélées par la projection comme des outliers ou des clusters. Cette extraction est un moyen de conserver des traces de l'analyse visuelle. Cependant, les techniques usuelles de sélection par manipulation directe comme le brossage (brushing) ne sont pas applicables directement aux projections. En effet à cause des artefacts de faux voisinages, elles ne garantissent pas à l'utilisateur que les groupes de points sélectionnés sont cohérents, c'est-à-dire qu'ils correspondent effectivement à des groupes d'individus similaires entre eux (cf. Figure 2.13).

Dans ce chapitre, nous proposons une nouvelle approche basée sur la métaphore de lentille pour visualiser interactivement les proximités d'origine. Cette lentille permet de déformer localement la projection, afin de la nettoyer de ses faux voisinages et ainsi permettre la mise en valeur des proximités d'origine dans le voisinage d'un individu de référence. Cette approche permet de créer une zone 2D sur la projection sans artefact de faux voisinages, ce qui rend possible le brossage des structures sous-jacentes aux données par manipulation directe sur la projection. Avant d'introduire les différents éléments de concept et de décrire notre exploration de l'espace de conception de cette lentille, nous rappelons dans cette section les problématiques rencontrées avec ProxiViz et introduisons la métaphore de lentille.

Problématiques

Les artefacts topologiques ont un impact sur la fiabilité des interprétations faites à partir de la projection pour les différentes tâches d'analyse visuelle (cf. section 2.4). En particulier, nous avons relevé que les projections avec principalement des artefacts de faux voisinages étaient moins faciles à interpréter que les projections présentant des artefacts de déchirures (cf. chapitre 4). Ces artefacts de faux voisinages nuisent au respect du voisinage d'origine dans le voisinage 2D d'un point de référence sur la projection. Mais ces erreurs locales de voisinage n'impactent pas seulement la projection. A cause de ces erreurs, ProxiViz souffre également de problèmes pour mettre en évidence le voisinage d'origine d'un individu de référence dans l'espace des données, ainsi que pour naviguer dans ce voisinage. Avec le problème de brossage sur la projection évoqué précédemment, l'aide à l'analyse visuelle avec ProxiViz pose les problématiques suivantes :

1. **Problème de représentation**

La représentation des proximités d'origine en utilisant une coloration interpolée peut souffrir de problèmes d'occlusion lorsque les points sont trop proches entre eux sur la projection, car cette coloration s'affiche en arrière plan du nuage de points. Ce phénomène d'occlusion devient pro-

blématique lorsqu'il s'ajoute au problème de faux voisinages. On ne peut alors plus distinguer les points qui sont réellement voisins de la référence dans l'espace des donnée, des points qui sont des faux voisinages (Figure 6.1).

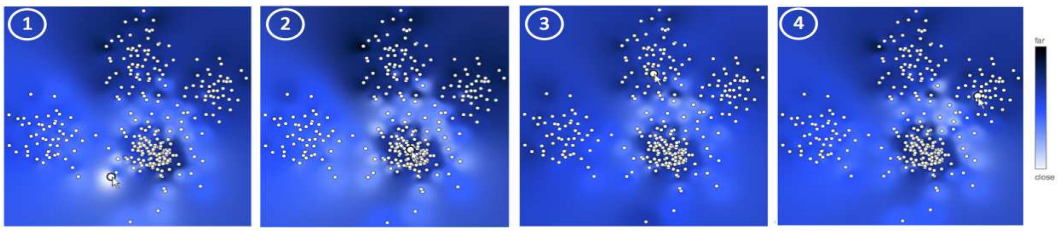


FIGURE 6.1 – ProxiViz souffre d'un problème d'occlusion : on ne distingue plus les points voisins de la référence dans l'espace des données. On remarque sur les différentes vues de ProxiViz (1-4) que l'encodage couleur change très faiblement. Ceci indique que les différents points de référence sont en réalité voisins dans l'espace des données. L'occlusion entre les points et les faux voisinages rendent ce voisinage d'origine difficile à deviner.

2. **Problème de navigation**

Nous avons déjà évoqué les problèmes de l'interaction de navigation liés à la sélection du point de référence (cf. chapitre 3). Nous avons introduit un système de délai dans la sélection du point de référence, mais celui-ci pose des problèmes relatifs à la vitesse de déplacement de la souris. Il nous faut trouver une solution plus robuste, car la sélection d'un faux voisinage équivaut à "téléporter" la référence dans une autre région de l'espace des données. Les proximités d'origine qui sont alors affichées sur la projection n'ont plus rien à voir avec les proximités par rapport à la référence précédente, ce qui implique une perte de contexte dans la navigation avec ProxiViz (Figure 6.2).

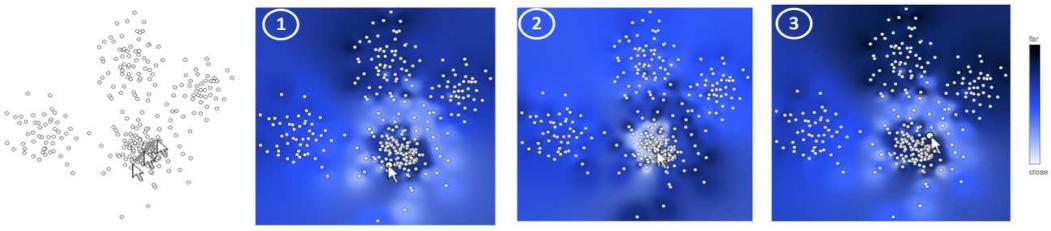


FIGURE 6.2 – ProxiViz souffre d'un problème d'interaction de navigation : la sélection de faux voisinages avec ProxiViz entraine une perte du contexte de navigation (1-3).

3. **Problème de brossage**

Les artefacts de faux voisinages entachent la "lecture" de la projection, mais ils sont également problématiques pour "l'extraction" d'information à partir de la projection. Des techniques de sélection sont nécessaires pour extraire les outliers ou clusters mis en évidence sur la projection et plus généralement pour conserver des traces de l'analyse visuelle. Mais les artefacts de faux voisinages nuisent au fonctionnement des techniques usuelles de sélection par manipulation directe.

Le brossage 2D [17] permet de mettre en valeur (ou masquer) des points sélectionnés par une interaction de *peignage*, c'est-à-dire lors du survol des points par une "brosse", représentée le plus souvent par un cercle (ou un carré) et dont la taille est ajustable. Cette technique est un moyen de faire des requêtes de filtrage dans les données par manipulation direct en "brossant" dynamiquement les objets à sélectionner. Dans le cas des projections, l'utilisateur, pensant sélectionner un groupe d'individus similaires en utilisant cette technique, peut être amené à sélectionner des faux voisinages, c'est-à-dire des points proches entre eux sur la projection mais très dissimilaires dans l'espace des données. L'utilisateur peut obtenir alors sans le savoir un groupe d'individus composé de plusieurs parties de clusters différents (Figure 6.3).

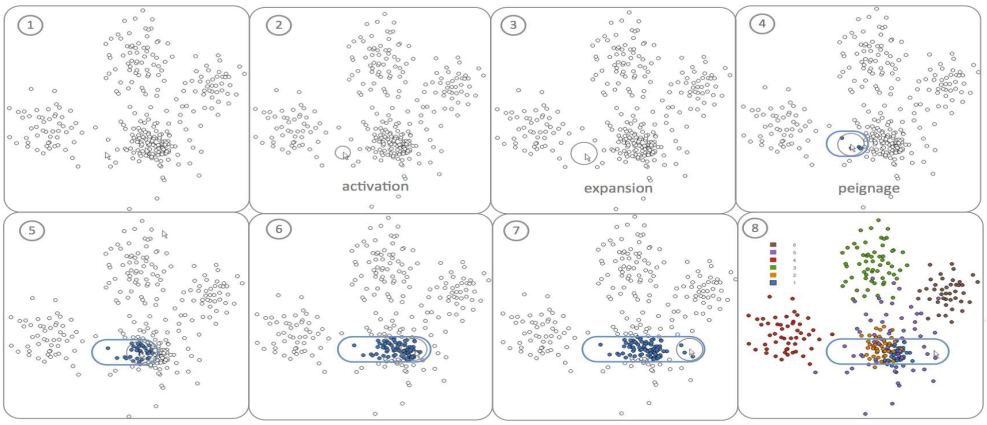


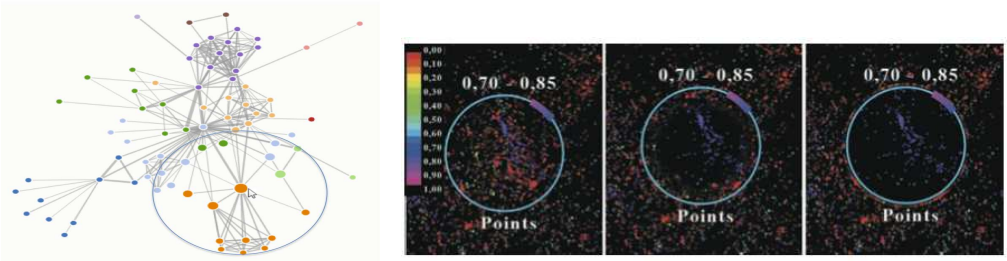
FIGURE 6.3 – Illustration du problème du brossage 2D sur la projection du jeu de données fil-rouge (1), pour lequel il existe un cluster sous-jacent à chaque classe (8). La brosse 2D correspond à un cercle (2), dont le diamètre est ajustable (3). L'interaction de peignage (4) permet de sélectionner les points lors du survol de la brosse (5,6,7). Mais le groupe d'individus résultant de ce brossage n'est pas cohérent à cause des artefacts de faux voisinages. En effet, les points de trois clusters différents sont sélectionnés. Ils sont donc regroupés ensembles alors qu'ils ne sont pas proches dans l'espace des données. Le cluster résultant de ce brossage n'existe pas dans l'espace des données.

Ainsi pour améliorer l'approche d'aide à l'analyse visuelle avec ProxiViz, nous proposons de résoudre ces problèmes en nettoyant localement la projection de ses faux voisinages par rapport à un point de référence donné. L'objectif étant de déformer la projection pour révéler interactivement les informations d'intérêt, c'est-à-dire les proximités d'origine relatives à la référence, cette approche suit le même principe que la métaphore de lentille.

Approche

En visualisation d'information, les lentilles [24] ou Fisheye [92], permettent de déformer localement la représentation pour dédier plus d'espace aux informations les plus importantes, c'est à dire déformer l'espace selon le degré d'intérêt des informations représentées (Figure 6.4). Différentes variations de cette technique existent pour les tables de données, les graphes ou les arbres. Parmi les variations de la lentille originale, MoleView [113] permet de filtrer des données relationnelles et multidimensionnelles selon un intervalle de valeurs sur une dimension donnée. Sur le graphe

des relations dans les données, la zone d'intérêt 2D contenue dans le périmètre de la lentille est nettoyée des points en dehors de l'intervalle d'intérêt, par une déformation de l'espace selon un champ de vecteurs. Ces points sont déplacés vers les bords de la lentille avec une animation suffisamment lisse pour que les utilisateurs conservent la perception du contexte, c'est-à-dire le modèle mental du positionnement initial.



(a) Lentille Fisheye (b) MoleView

FIGURE 6.4 – Lentille Fisheye permettant de magnifier une région du graphe (a) et MoleView appliqué à la projection MDS d'un corpus de documents (b). On remarque pour MoleView que les points en dehors de l'intervalle d'intérêt sont poussés vers les bords de la lentille.

Notre approche couple la technique ProxiViz avec le concept de MoleView et applique cette métaphore de lentille au filtrage des artefacts de faux voisinages. Dans ce chapitre, nous introduisons l'espace de conception associé à cette lentille dans le but de résoudre les problèmes de ProxiViz. Cette espace de conception repose sur une définition des artefacts topologiques pour laquelle nous introduisons deux éléments de concept : un opérateur de sélection dans l'espace des données et un opérateur de filtrage dans l'espace de la projection. A partir de cette définition abstraite, nous introduisons une taxonomie des points par rapport à la référence de ProxiViz. La représentation de cette taxonomie revient à construire la lentille : elle permet de filtrer visuellement les faux voisinages pour mettre en évidence la coloration des proximités d'origine dans le voisinage 2D de la référence. Nous discutons ensuite la prise en compte de cette taxonomie dans l'interaction de navigation et l'interaction de brossage sur la projection. Finalement, nous introduisons ProxiLens, une implémentation de cette lentille combinant deux opérateurs de voisinage afin d'aider à effectuer des tâches d'analyse locale et globale. Nous illustrons cette technique et ses paramètres sur les différentes tâches d'analyse visuelle avec un jeu de données d'images. Puis nous discutons les enjeux d'amélioration de cette approche.

6.2 Espace de conception

L'espace de conception de la lentille est génératif. Il existe une multitude de possibilités et de combinaisons des différents opérateurs, représentations et interactions décrites dans la suite. Notre démarche consiste à définir des concepts et critères permettant de catégoriser et comparer les différentes possibilités afin de résoudre les problèmes énoncés précédemment. Nous présentons également différentes combinaisons que nous avons implémentées.

6.2.1 Conceptualisation de la lentille

Pour construire une lentille filtrant les artefacts de faux voisinages, nous devons revenir à la définition des artefacts topologiques (cf. section 2.3). Cette définition est également utilisée pour projeter des données [245] et elle repose sur la comparaison entre le voisinage d'une référence dans l'espace des données et le voisinage du point correspondant à la référence dans l'espace de projection.

Les artefacts topologiques ont deux caractéristiques importantes : relativité et granularité. On peut considérer un point (ou un ensemble de points) comme étant un artefact topologique relativement à un autre point (ou ensemble de points). Par exemple, si on considère un cluster qui a été découpé en plusieurs composantes non connexes sur une projection, chaque composante est alors un artefact de déchirure du point de vue des autres composantes, mais elles sont également potentiellement des faux voisinages pour les clusters à proximité desquels elles ont été projetées.

Pour définir les artefacts topologiques, on doit considérer un voisinage relatif à une référence x dans l'espace des données ainsi qu'un voisinage relatif au point y représentant la référence dans l'espace 2D. Afin de prendre en compte la notion de granularité, nous introduisons deux opérateurs permettant d'abstraire ces voisinages (Figure 6.5) :

> *Opérateur de sélection* : cet opérateur, noté σ, correspond à l'application qui associe un ensemble d'individus similaires à une référence, pour une valeur seuil donnée α définissant l'étendue du voisinage autour de cette référence dans l'espace des données.
>
> *Opérateur de filtrage* : cet opérateur, noté ϕ, correspond à l'application qui associe un ensemble de points à une référence, pour une valeur seuil donnée β définissant l'étendue du voisinage autour de cette référence dans l'espace de projection.

Le voisinage défini par l'opérateur de filtrage dans l'espace de projection correspond, dans la ensuite, à la zone de filtrage 2D de la lentille. Le voisinage définit par l'opérateur de sélection dans l'espace des données correspond lui à la sélection des individus proches de la référence pour lesquels on souhaite visualiser et analyser les proximités d'origine.

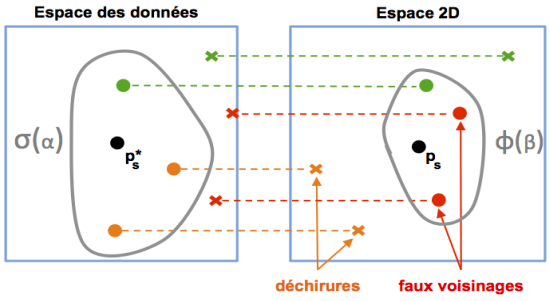


FIGURE 6.5 – Schéma illustrant les artefacts topologiques : faux voisinages et déchirures. Ces artefacts relativement à l'individu x correspondent à des erreurs de préservation entre le voisinage d'étendue α dans l'espace des données et le voisinage d'étendue β dans l'espace de projection 2D. Les points verts sont présents (rond) ou absents (croix) du voisinage dans les deux espaces.

Taxonomie des points relative à la référence

On peut voir le processus de projection comme un processus de classification où le voisinage dans l'espace des données définit la classe des voisins et celle des non-voisins, c'est-à-dire les classes réelles ; et le voisinage dans l'espace de projection définit le résultat de la classification, c'est-à-dire les classes estimées. A l'image d'une tâche de classification automatique, on peut définir une matrice de confusion. En comparant le voisinage d'origine dans l'espace des données avec le voisinage dans l'espace 2D relativement à un point de référence sélectionné, on obtient pour chaque point de la projection un des 4 cas suivants (Figure 6.6) :

Vrai Positif $\sim$ *Vrai Voisin :* Point bien classé ("hit"), c'est-à-dire un point qui est voisin de la référence à la fois dans l'espace 2D et dans l'espace des données.

Faux Positif $\sim$ *Faux Voisin :* Point d'une autre classe mal classé ("false alarm"), c'est-à-dire un point qui est voisin de la référence dans l'espace 2D mais pas dans l'espace des données.

Faux Négatif $\sim$ *Déchirure :* Point de la classe mal classé ("miss"), c'est-à-dire un point qui est voisin de la référence dans l'espace des données mais pas dans l'espace 2D.

Vrai Négatif $\sim$ *Non Voisin :* Point d'une autre classe bien classé ("correct rejection"), c'est-à-dire un point qui n'est pas voisin de la référence dans les deux espaces.

		Espace des données	
		Voisin	Non-Voisin
Espace de projection	Voisin	Vrai Voisin	Faux Voisin
	Non-Voisin	Déchirure	Non-Voisin

FIGURE 6.6 – Matrice de confusion du processus de projection, c'est-à-dire la taxonomie des points relative à la référence.

Après une présentation des opérateurs de sélection et de filtrage, nous introduisons la représentation graphique de la taxonomie des points relative à la référence de ProxiViz. Cette représentation permet de construire une lentille qui nettoie localement la projection de ses faux voisinages afin de résoudre les problématiques de ProxiViz, aussi bien au niveau de la mise en évidence des proximités d'origine, que de l'interaction de navigation et de brossage sur la projection.

Opérateurs de sélection

Comme nous l'avons vu dans la section dédiée aux algorithmes de projection, différentes approches existent pour définir un voisinage dans l'espace des données. Par exemple, dans l'Analyse en Composantes Curvilignes, le voisinage est défini par seuillage des proximités entre les individus en fonction d'un scalaire défini en paramètre de l'algorithme. Dans Isomap, le voisinage est défini par la valeur k du rang définissant le graphe des *k*-plus proches voisins. La définition d'un voisinage par seuillage dans l'espace des données est également utilisée dans les algorithmes de clustering automatiques pour agréger les individus par densité comme DBSCAN [77] ou par classification ascendante hiérarchique (CAH) [164].

Nous ne nous intéressons pas ici à la définition d'un voisinage théorique mais plutôt à un moyen de définir une région dans l'espace des données. On peut envisager un grand nombre d'opérateurs possibles pour sélectionner des individus en fonction de leur proximité à un individu référence dans l'espace des données. Nous considérons ici trois opérateurs de sélection afin d'illustrer ce concept (Figure 6.7) :

Géométrique : cet opérateur retourne l'ensemble des individus compris dans le voisinage géométrique de la référence x_i, c'est-à-dire à une distance d inférieure à la valeur seuil α : $\sigma_\alpha(x_i) = \{x_k \in \mathbb{R}^{n\times m} | d(x_i,x_k) \leq \alpha\}$. Pour un individu référence, cet opérateur peut être vu comme l'hypersphère centré sur l'individu et dont le rayon correspond à la valeur seuil α.

Géodésique : cet opérateur retourne l'ensemble des individus compris dans le voisinage géodésique de la référence x_i, c'est-à-dire à une distance géodésique d_G inférieure à la valeur seuil α sur le graphe G des k-plus proches voisins de la référence : $\sigma_\alpha(x_i) = \{x_k \in \mathbb{R}^{n\times m} | d_G(x_i,x_k) \leq \alpha\}$. Cet opérateur permet de visualiser les connections de proximité entre individus à la manière de Spider Cursor [225].

Agrégatif : cet opérateur retourne l'ensemble des individus $\{x_1,...,x_k\}$ appartenant à un cluster C_α englobant la référencex_i et dont le nombre de points est fonction de α : $\sigma_\alpha(x_i) = \{x_k \in \mathbb{R}^{n\times m} | (x_i,x_k) \in C_\alpha\}$. Cet opérateur considère des algorithmes de clustering qui fournissent une partition totale des données et pour lesquels on peut obtenir un cluster de taille différente en fonction de α, relativement à un individu référence. La CAH est un bon candidat pour cet opérateur car elle fournit un dendrogramme, c'est-à-dire une hiérarchie de clusters. La hauteur dans le dendrogramme paramétre directement le nombre de points dans le cluster. Aussi on peut associer α à la hauteur de coupure dans le sous-arbre contenant la référence.

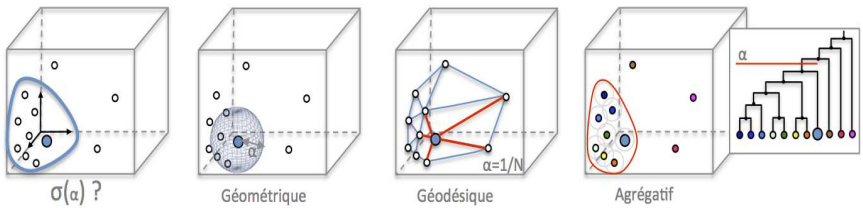


FIGURE 6.7 – Exemple d'opérateurs de sélection définissant le voisinage de la référence dans l'espace des données.

L'étendue du voisinage dans l'espace des données, contrôlée par α, peut s'exprimer en termes du nombre d'individus N_α dans le voisinage, c'est-à-dire $f(\alpha) = |N_\alpha|$ avec $\alpha \in [0,1]$. Le voisinage est vide pour $\alpha = 0$ et il contient tous les individus pour $\alpha = 1$ (ce qui correspond à ProxiViz). Ce paramètre d'étendue étant borné, il peut être contrôlé à l'aide d'un curseur de défilement (slider). Mais l'évolution de l'étendue dépend de l'opérateur de sélection et n'est pas obligatoirement linéaire. Pour un opérateur de sélection *agrégatif*, par exemple, à chaque incrément/décrément de α, il y a un nombre variable d'individus qui sont ajoutés/retirés du voisinage.

L'étendue du voisinage, en termes de nombre de points, peut varier d'une référence à une autre pour une même valeur α. En effet, si on considère l'opérateur de sélection *géométrique*, pour une valeur α, le voisinage contiendra beaucoup plus de points dans des régions denses de l'espace des données que dans des régions moins denses. Un réglage automatique de α en fonction de la densité est envisageable mais l'utilisateur ne sera alors plus conscient de la distribution des proximités dans l'espace des données. Aussi nous préférons considérer une valeur seuil α fixée par l'utilisateur, afin qu'il puisse appréhender la densité dans l'espace des données en adaptant manuellement l'étendue du voisinage.

De ces trois opérateurs de sélection, l'opérateur *géométrique* semble le plus intuitif, car il correspond à un simple seuillage des proximités d'origine relatives à la référence. Cependant comme il centre le voisinage sur la référence dans l'espace des données, il rend la sélection de tout un cluster assez difficile à partir d'une unique référence. L'opérateur de sélection *géodésique* pose le même problème, mais étant moins facile à se représenter mentalement nous ne considérons pas cet opérateur dans la suite. L'opérateur de sélection *agrégatif*, en revanche, est tout indiqué pour sélectionner un cluster, mais il pose le problème du choix du critère d'agrégation. Par exemple pour la CAH, l'agrégation peut se faire selon différents critères de liaison en fonction de la distance minimum (liaison simple) ou maximum (liaison complète) sur l'ensemble des individus du cluster. Nous utilisons dans la suite, une CAH avec un critère de liaison simple pour que le clustering soit comparable avec celui que l'on pourrait réaliser manuellement en explorant l'espace des données avec la lentille munie de l'opérateur de sélection *géométrique*, c'est-à-dire par agrégation itérative de sphères de rayon α dans l'espace des données.

Opérateurs de filtrage

L'objectif de la lentille est de nettoyer localement la projection de ses faux voisinages relativement à un point de référence. Pour cela, il faut définir une zone représentant le voisinage de la référence dans l'espace 2D, que nous appellerons zone de filtrage. Cette zone 2D correspond d'abord à un moyen de résoudre les problèmes de ProxiViz avant de correspondre à un voisinage théorique. En fonction du voisinage dans l'espace des données qui correspond à un ensemble d'individus sélectionnés, nous distinguons deux catégories d'opérateurs de filtrage :

Indépendant : Cette catégorie d'opérateurs est indépendante de l'ensemble des individus sélectionnés et correspond à des opérateurs qui définissent une zone 2D fixée autour de la référence dont l'étendue est réglée par la valeur du seuil β. Un cercle de rayon β centré sur le point de référence est un exemple d'opérateur appartenant à cette catégorie.

Dépendant : Cette catégorie d'opérateurs est dépendante de l'ensemble des individus sélectionnés et adapte le voisinage 2D en définissant un périmètre de filtrage, dont l'étendue est réglée par valeur du seuil β, autour (ou contenant) chaque point correspondant à un individu sélectionné. L'enveloppe convexe de ces points ou leur alpha shape [71], avec une marge β, sont des exemples d'opérateurs appartenant à cette catégorie. D'autres variantes existent comme les enveloppes papillons [193] ou la représentation de l'arbre couvrant minimum basé sur le graphe de Delaunay 2D entre les points correspondant aux individus sélectionnés.

On peut définir des opérateurs semblables à ceux utilisés dans l'espace des données pour chacune des deux catégories (Figure 6.8) : géométrique, géodésique, agrégatif. L'opérateur de filtrage *géométrique indépendant* se définit ainsi comme suit : $\phi_\beta(y_i) = \{y_k \in \mathbb{R}^{n\times 2} | d(y_i, y_k) \leq \beta\}$. Pour un point référence, cet opérateur peut être vu comme le cercle centré sur le point et dont rayon correspond à la valeur seuil β. L'opérateur de filtrage *géométrique dépendant* revient à définir un opérateur de filtrage *géométrique* en chaque point correspondant à un individu sélectionné, et se définit comme suit : $\phi_\beta(y_i) = \{y_k \in \mathbb{R}^{n\times 2} | \forall y_j \in \sigma(x_i), d(y_j, y_k) \leq \beta\}$. Ces deux opérateurs sont illustrés dans la section suivante sur la représentation de la zone de filtrage 2D.

L'opérateur de filtrage *géométrique* est intuitif en 2D, car on peut représenter les cercles utilisés pour le filtrage et ainsi visuellement se faire une idée de la valeur du paramètre β. Le paramètre β est directement lié à l'étendue de la zone 2D, laquelle dépend du rayon du cercle (ou des cercles) de filtrage, ou bien de l'épaisseur de la marge de l'enveloppe convexe autour des points. Ainsi ce paramètre de filtrage β s'exprime en nombre de pixels et il est borné par le diamètre de la visualisation de la projection. Ce paramètre étant borné, il peut être contrôlé à l'aide d'un curseur de défilement (slider). Pour $\beta = 0$, aucun filtrage n'est opéré sur les artefacts topologiques et pour une valeur β égale au diamètre de la projection, seuls les points correspondant à des individus sélectionnés ne sont pas filtrés sur la projection.

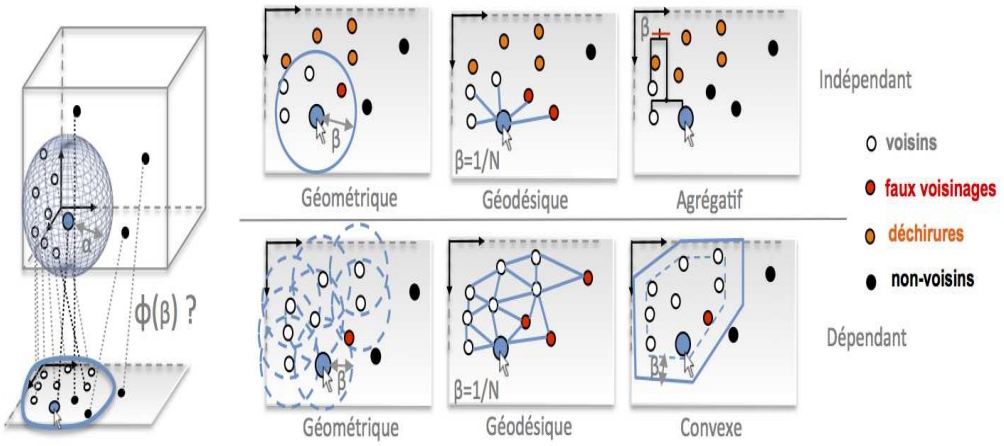


FIGURE 6.8 – Exemple d'opérateurs de filtrage définissant le voisinage de la référence dans l'espace de projection.

Une fois les opérateurs de sélection et de filtrage définis, la représentation de la taxonomie des points définie précédemment permet de construire une lentille qui nettoie localement la projection de ses faux voisinages. Les sous-sections suivantes présentent les enjeux de représentation et d'interaction associés à cette lentille et montrent en quoi cette approche résout les problèmes de ProxiViz en termes de mise en évidence des proximités d'origine, d'interaction de navigation et de brossage sur la projection.

6.2.2 Représentation de la lentille

La représentation de la lentille est une adaptation de l'encodage couleur de ProxiViz reposant sur la taxonomie des points relative à une référence. Un des objectifs de cette représentation est de résoudre les problèmes d'occlusion de ProxiViz associés au faux voisinages (cf. section 6.1). Nous cherchons donc un encodage visuel de la taxonomie qui mette en évidence la coloration des proximités d'origine des points proches de la référence.

Différentes techniques existent pour réduire l'occlusion dans une visualisation [73], parmi lesquelles le changement de la taille, de la transparence, ou de la position. Pour rendre ces modifications d'encodage plus agréables, nous utilisons une animation de transition (Figure 6.11). Nous avons considéré les encodages suivants pour "filtrer" les faux voisinages :

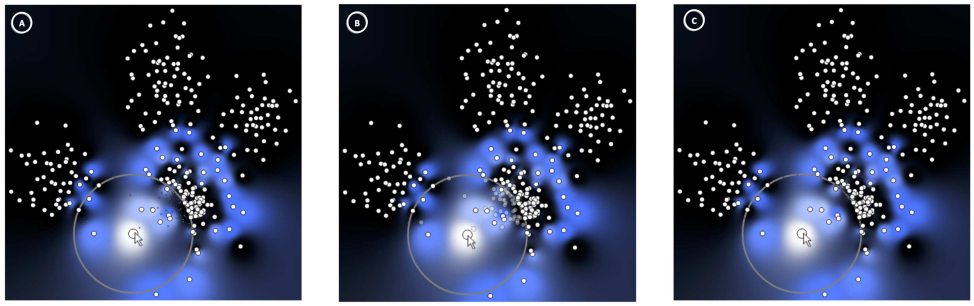
Re-projection locale des voisins d'origine : cet encodage optimise la position des points sélectionnés (vrais voisins et déchirures) par re-projection dans la zone de filtrage 2D et repousse les faux voisinages vers les bords de cette zone (Figure 6.10). Cet encodage impacte le contexte de la projection, car il déforme également la projection en dehors de la zone de filtrage lorsqu'il déplace les déchirures. La nouvelle projection locale doit rester cohérente en termes de respect des proximités d'origine avec le contexte autour de la zone de filtrage (il faut contraindre les rotations de la re-projection). Cependant selon de l'envergure de la zone de filtrage, cette re-projection peut souffrir de problèmes de proportions introduisant des artefacts géométriques. De plus, si l'animation du déplacement permet de suivre les points, celle-ci rend l'interaction plus complexe, car on doit attendre que la position des points soit stabilisée pour sélectionner une nouvelle référence.

Déplacement des faux voisinages vers les bords de la zone de filtrage 2D : cet encodage déforme la projection dans le périmètre de la zone de filtrage (Figure 6.9-C et Figure 6.13). Même si l'animation du déplacement permet de suivre les points, cet encodage fait perdre le contexte initial de la projection, c'est-à-dire que l'on ne sait plus où étaient les points initialement. Par exemple, on ne peut plus compter combien de faux voisinages étaient présents dans la zone de filtrage. De plus, l'attention de l'utilisateur est naturellement captée par le déplacement des points, plutôt que par les informations mises en évidence dans la zone de filtrage.

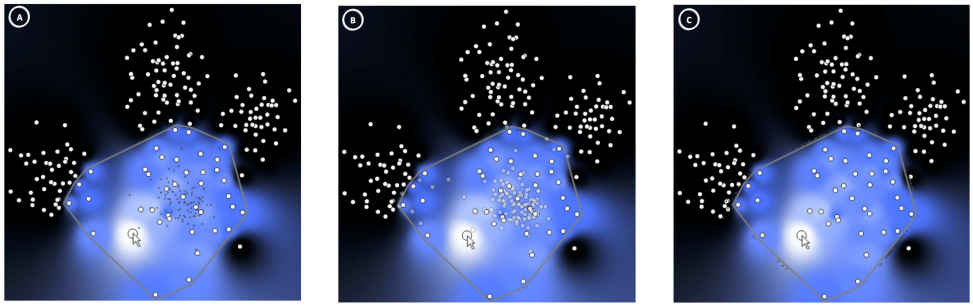
Mise en transparence des faux voisinages : cet encodage préserve également le contexte de la projection, c'est-à-dire que la position des points reste inchangées, mais il ne réduit pas nettement le phénomène d'occlusion (Figure 6.9-B). En revanche, il permet toujours de distinguer des différences entre les étiquettes de classe des points.

Réduction de la taille des faux voisinages : cet encodage préserve le contexte de la projection et il réduit très nettement la visibilité des faux voisinages (Figure 6.9-A). Nous choisissons cet encodage dans la suite, même si en contre partie, il ne permet plus de distinguer les éventuelles étiquettes de classe sur les faux voisinages.

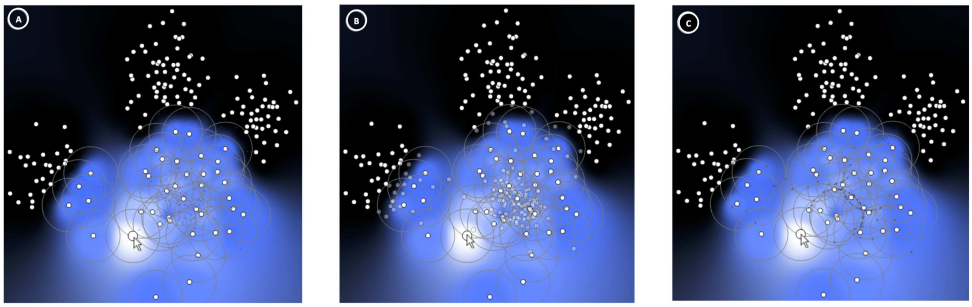
Le critère permettant de choisir un encodage de "filtrage" des faux voisinages repose sur l'équilibre entre la déformation de l'encodage visuel de la projection induite par ce filtrage et la mise en évidence des proximités dans le voisinage d'origine de la référence sur la zone 2D de filtrage, c'est-à-dire la réduction du phénomène d'occlusion par ces faux voisinages (Figure 6.13).



(a) Opérateur de filtrage *géométrique indépendant*

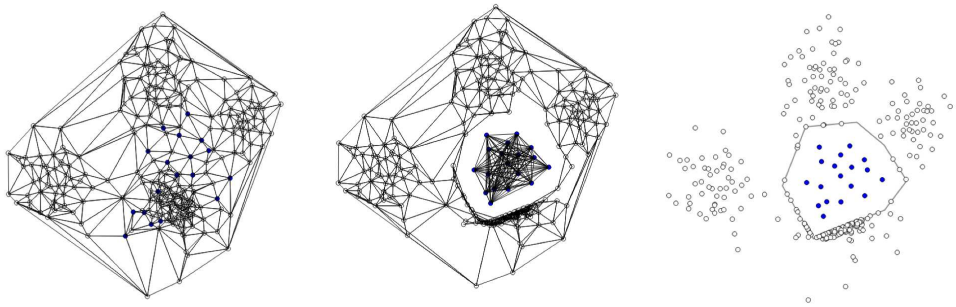


(b) Opérateur de filtrage *enveloppe convexe dépendante*



(c) Opérateur de filtrage *géométrique dépendant*

FIGURE 6.9 – Exemples d'encodages visuels de la taxonomie des points pour filtrer les faux voisinages. Nous considérons les opérateurs de filtrage : *géométrique indépendant* (a), *enveloppe convexe dépendante* (b), *géométrique dépendant* (c). Ainsi que les encodages des faux voisinages suivants : réduction de la taille (A), mise en transparence (B), déplacement vers les bords (C), radialement vers les bords des cercles ou orthogonalement vers la frontière la plus proche de l'enveloppe convexe. L'interpolation de la coloration des proximités d'origine est restreinte au voisinage d'origine de la référence (par défaut une couleur noir est appliquée aux non-voisins dans l'interpolation). Nous considérons les opérateurs de sélection *géométrique* ($\alpha = 0.4$) et nous affichons les bords de la zone de filtrage 2D avec un contour gris.



(a) Graphe de Delaunay 2D (b) Optimisation du sous graphe des individus sélectionnés (c) Résultat de la re-projection

FIGURE 6.10 – Re-projection locale des individus sélectionnés (en bleu) en utilisant une approche de placement par force basée sur le graphe de Delaunay 2D. Nous considérons pour cet exemple un opérateur de filtrage avec une enveloppe convexe, afin que les points re-projeté puissent prendre la place nécessaire pour ne pas introduire de nouveaux artefacts. La marge de l'enveloppe convexe contenant les points est réglable selon β afin de trouver un équilibre entre la mise en évidence des proximités dans le voisinage d'origine et la déformation de la projection. L'approche de placement par force utilise les arrêtes du graphe de Delaunay 2D et permet de repousser les faux voisinages en dehors de l'enveloppe convexe, sans trop déformer le reste de la projection. Nous n'affichons pas la coloration de Proxiviz, car la re-projection dans cet exemple permet de respecter les proximités d'origine.

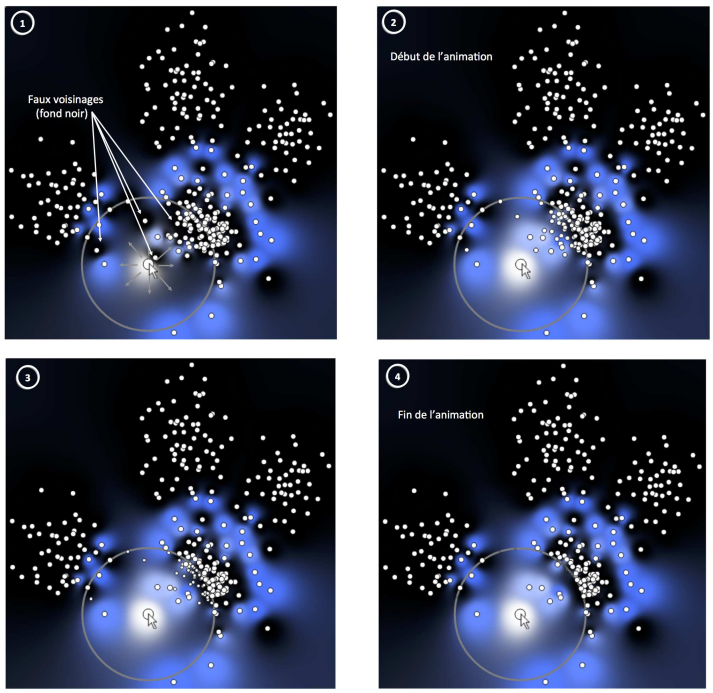


FIGURE 6.11 – Animation du déplacement des faux voisinages vers les bords de la lentille. Nous considérons ici un opérateur de sélection *géométrique* ($\alpha = 0.4$) et un opérateur de filtrage *géométrique indépendant* ($\beta = 100 pixels$).

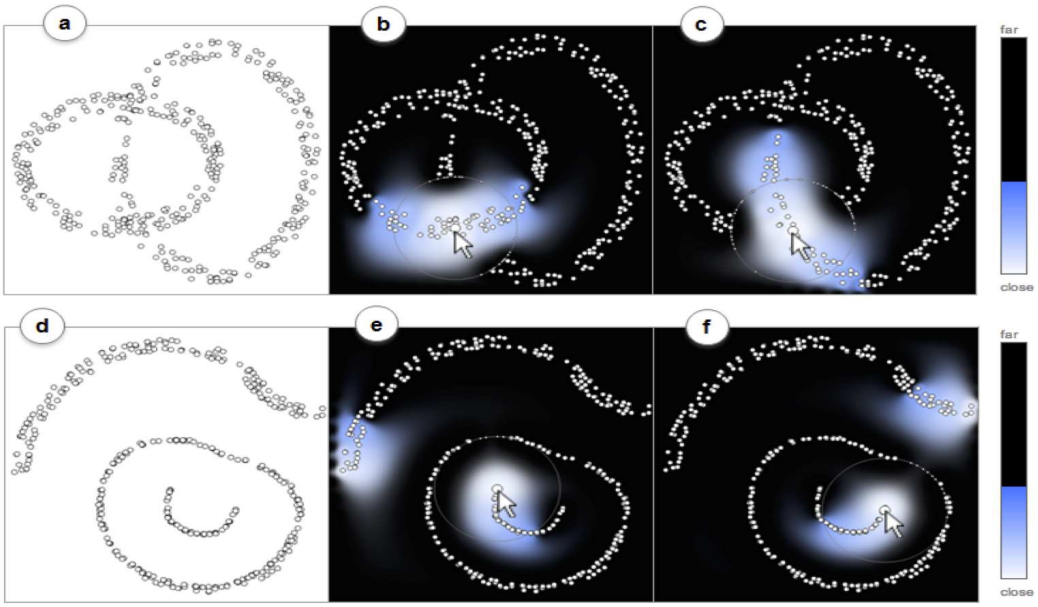
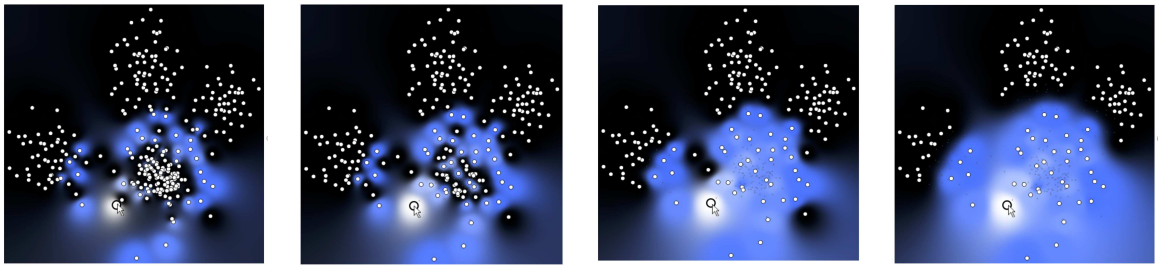


FIGURE 6.12 – Exemple de lentille appliquée à deux anneaux entrelacés en 3D positionnés un plan perpendiculaire à l'autre et projetés en 2D par une Analyse en Composantes Principales (a-b-c) et une Analyse en Composantes Curvilignes (d-e-f). Dans cette configuration, les opérateurs de sélection et de filtrage sont *géométriques* (avec $\alpha = 0.2$ et $\beta = 100$pixels), avec un déplacement des faux voisinages vers les bords de la lentille comme dans la technique MoleView [113]. La lentille permet une navigation continue dans l'espace des données le long de l'anneau pointé malgrés les faux voisinages (b-c) et les déchirures (e-f).



(a) $\beta = 0px$ (b) $\beta = 75px$ (c) $\beta = 150px$ (d) $\beta = 300px$

FIGURE 6.13 – Réglage manuel du paramètre β de l'opérateur de filtrage pour optimiser le confort d'utilisation de la lentille. Avec l'augmentation de β, les proximités d'origine sont mieux mises en valeur, mais ceci au détriment du contexte de la projection. Nous considérons ici un opérateur de sélection *géométrique* ($\alpha = 0.4$) et un opérateur de filtrage *géométrique dépendant*.

Par exemple, l'utilisation de l'enveloppe convexe comme opérateur de filtrage ne minimise pas la déformation, car elle constitue une zone de filtrage importante et arbitraire selon la position des individus sélectionnés sur la projection. A l'inverse, l'utilisation d'une zone de filtrage circulaire autour du point de référence est trop localisée et ne met pas en valeur les déchirures parmi les individus sélectionnés.

En revanche, l'utilisation d'un cercle autour de chaque points, correspondant à des individus sélectionnés, minimise la déformation globale de la projection pour préserver le contexte et met clairement en évidence le voisinage d'origine de la référence sans problèmes d'occlusions. Cet opérateur de filtrage *géométrique dépendant*, proche de l'alpha shape, est en particulier utile pour contrôler les points qui contribuent à l'interpolation de la coloration et permet ainsi de contrôler l'efficacité de la mise en évidence des proximités d'origine relatives à la référence. Nous retenons donc cet encodage avec une réduction de la taille des points pour résoudre les problèmes d'occlusion de ProxiViz et représenter la lentille dans la suite (Figure 6.13).

Afin de mettre en évidence les individus sélectionnés, c'est-à-dire ceux qui appartiennent au voisinage d'origine de la référence, nous ne colorons pas les proximités d'origine des autres points, c'est-à-dire les non-voisins et les faux voisinages. Nous pourrions utiliser deux teintes de couleur, l'une pour les points voisins de la référence dans l'espace des données et l'autre pour les points non-voisins à l'origine, afin d'afficher les proximités d'origine sur toute la projection, comme avec ProxiViz. Mais le filtrage des faux voisinages implique de ne pas prendre en compte ces points dans l'interpolation de la couleur afin de mieux visualiser les informations relatives aux points d'intérêt, c'est-à-dire les points à l'intérieur de la lentille. Or selon l'opérateur de sélection utilisé, ces faux voisinages peuvent être plus proche de la référence dans l'espace des données que les autres points en dehors du voisinage d'origine de la référence. Ainsi la coloration des proximités d'origine en dehors de la lentille n'est pas représentative de la réalité dans l'espace des données. Nous décidons, par conséquent, de ne pas colorer les faux voisinages et d'appliquer par défaut une couleur noir aux non-voisins dans l'interpolation de la coloration des proximités d'origine.

Il est a noté que l'importance de la déformation de la projection dépend également des opérateurs de sélection et de filtrage. En effet, l'amplitude de la déformation est liée l'envergure de la zone de filtrage 2D, paramétrée par β, ainsi qu'au nombre de faux voisinages dans cette zone qui lui dépend de l'étendue du voisinage d'origine paramétré par α. La sous-section suivante discute l'interaction avec ces opérateurs après avoir montré en quoi la lentille permet de résoudre les problèmes d'interaction de navigation avec ProxiViz et de brossage sur la projection.

6.2.3 Interaction avec la lentille

L'interaction avec la lentille concerne le paramétrage des opérateurs de sélection et de filtrage, mais également la sélection de la référence, c'est-à-dire l'interaction de navigation et le brossage par peignage pour conserver des traces de l'analyse visuelle. Ces deux derniers points correspondent à des problèmes de ProxiViz associés au faux voisinages (cf. section 6.1). Nous cherchons donc à les résoudre en prenant en compte la taxonomie des points relative à la référence dans les interactions avec la lentille.

Interaction de navigation

La référence est sélectionnable directement sur la projection par clic ou automatiquement par survol des points. L'utilisation des cellules de Voronoï permet de sélectionner automatiquement le point le plus proche du curseur de la souris. Cependant la sélection de la référence est rendue difficile par la présence de faux voisinages. En effet, la sélection d'un faux voisin équivaut à "téléporter" la référence dans une autre région de l'espace des données (Figure 6.15). Aussi les individus sélectionnés ne seront pas voisins dans l'espace des données des individus sélectionnés précédemment, chose que nous voulons éviter, car cela perturbe l'analyse visuelle des proximités d'origine affichées par ProxiViz. Pour résoudre ce problème de navigation, il faut donc prendre en compte les faux voisinages afin de faire la différence entre les points potentiellement sélectionnables et ceux devant être évités.

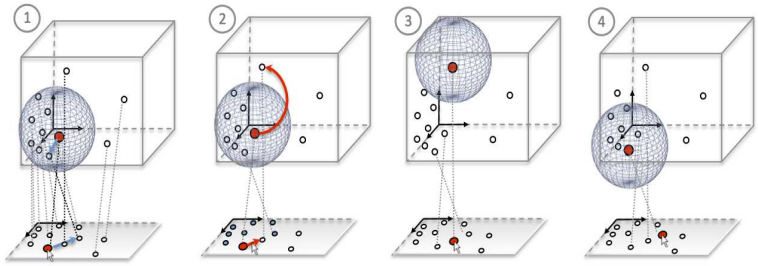


FIGURE 6.14 – Exemple de "téléportation" illustrant l'impact des faux voisinages sur l'interaction de navigation avec la lentille. Nous considérons ici un opérateur de sélection *géométrique* pour définir le voisinage d'origine dans l'espace des données. On remarque qu'en explorant deux points voisins sur la projection, la nouvelle référence n'est pas voisine de la précédente dans l'espace des données (1-4). Nous proposons donc d'inhiber la sélection des faux voisinages comme références.

Nous proposons d'inhiber la sélection des faux voisinages comme références, afin que l'utilisateur puisse naviguer l'espace des données de manière continue, c'est-à-dire visualiser les proximités d'origine d'individus voisins en individus voisins dans l'espace des données. Ceci revient à construire un *espace de navigation* dans lequel l'utilisateur peut explorer de manière continue l'espace des données (Figure 6.15). Dans cet espace de navigation, l'utilisateur peut choisir une nouvelle référence parmi les points voisins sans problématiques de faux voisinages. Nous considérons dans la suite que cet espace de navigation est accessible comme un *mode d'interaction*, que l'utilisateur peut activer/désactiver lorsqu'il explore la projection. La désactivation de ce mode, que nous appellerons *mode de navigation*, permet de sélectionner des faux voisinages sur demande de l'utilisateur pour naviguer sur des outliers de données.

Avec l'utilisation d'une sélection automatique de la référence basée sur les cellules de Voronoï, l'utilisateur peut être amené à sélectionner des références intermédiaires lorsqu'il souhaite explorer une déchirure, qui serait séparée du point de référence courant par des points non-voisins. Afin de résoudre ce problème qui rompt le continuum de la navigation, nous proposons d'introduire un autre *mode d'interaction*, dans lequel la sélection automatique de la référence par proximité au curseur de la souris est temporairement inhibée jusqu'à la désactivation du mode. Dans ce mode d'interaction, que nous appellerons *mode de sélection*, ProxiViz est figé et l'utilisateur peut librement choisir une nouvelle référence parmi les individus sélectionnés, avant de désactiver le mode

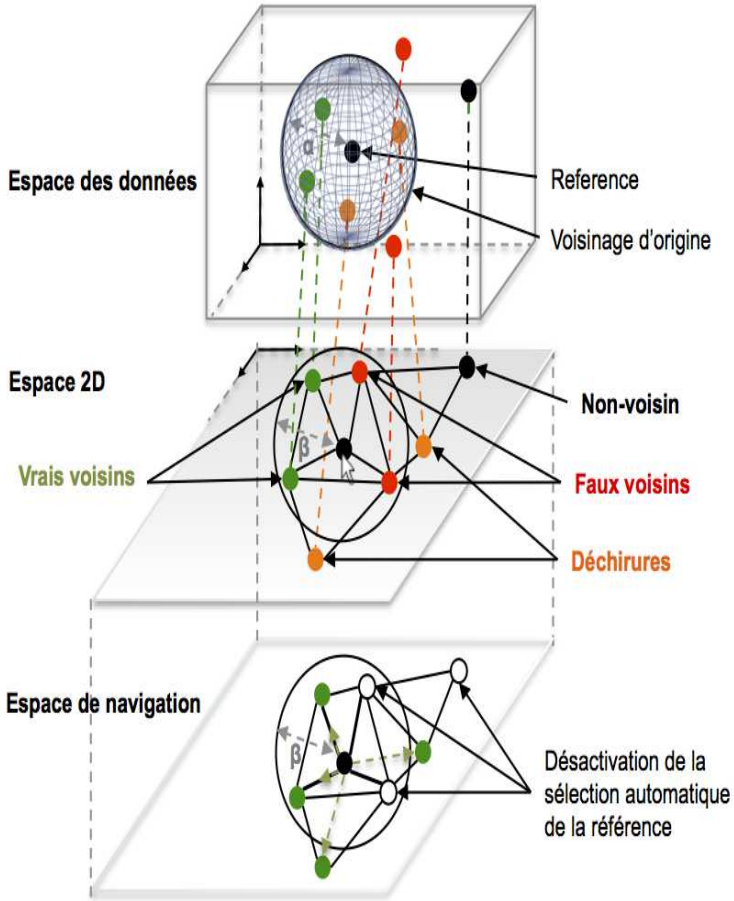


FIGURE 6.15 – Espace de navigation. Pour définir cet espace, on utilise la taxonomie des points relative à la référence (en noir) avec des *opérateurs de sélection et filtrage géométriques*. La sphère centrée sur la référence avec pour rayon α définit le voisinage d'origine dans l'espace des données (cercle au contour noir). Les vrais voisins et les non-voisins sont colorés en vert. Les artefacts sont les points qui sont dans le voisinage de la référence dans un espace mais pas dans l'autre. On distingue deux types d'artefacts topologiques : les faux voisinages (en rouge), qui sont des individus absents du voisinage d'origine de la référence dans l'espace des données mais qui sont représentés sur la projection par des points dans le voisinage 2D du point de référence ; les déchirures (en orange) qui sont les individus présents dans le voisinage d'origine dans l'espace des données mais absents du voisinage 2D du point de référence sur la projection. L'espace de navigation correspond au voisinage d'origine de la référence. Dans l'espace de navigation, seuls les vrais voisins et les déchirures peuvent être explorés (flèches vertes en pointillés) en préservant une continuité dans la navigation de l'espace des données. Les faux voisinages et les non-voisins ne peuvent pas être sélectionnés comme références. Les arrêtes noires relient des voisins dans le graphe de Delaunay 2D et indiquent quels points voisins de la référence sont sélectionnables automatiquement selon leur proximité 2D au pointeur de la souris.

et de retrouver une coloration interactive des proximités d'origine par rapport à la référence sélectionnée automatiquement selon la proximité du curseur de la souris. Ce mode étant temporaire, il peut être activé par pression d'une touche (ou d'un bouton de la souris) et désactivé au relâchement de la touche (ou du bouton).

Pour mettre en évidence ce *mode de sélection* sur la projection, nous proposons d'adapter temporairement l'opérateur de filtrage et d'utiliser l'enveloppe convexe des points sélectionnés, avec une marge β (Figure 6.16). En effet, dans ce mode l'objectif est de clairement mettre en évidence les individus dans le voisinage d'origine de la référence, c'est-à-dire les potentielles références sélectionnables par l'utilisateur.

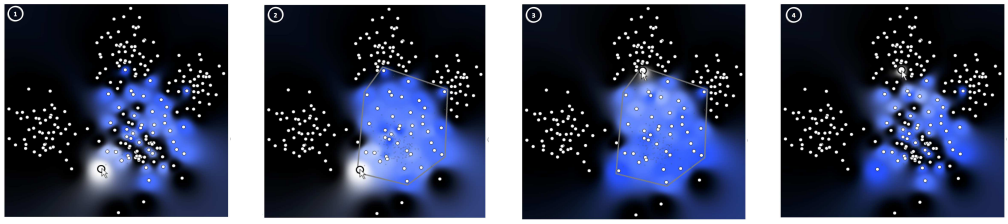


FIGURE 6.16 – Interaction dans le *mode de sélection*. Sur activation de ce mode (1), la représentation change pour mettre en valeur les voisins d'origine de la référence (2). La sélection automatique est inhibée pour permettre de sélectionner la prochaine référence sans le problème des artefacts (3). En déasctivant le *mode de sélection*, on peut reprendre la navigation dans une nouvelle région de la projection, mais toujours dans la même région de l'espace des données (4). Nous considérons ici un opérateur de sélection *agrégatif* et un opérateur de filtrage *géométrique dépendant*.

Interaction de brossage

Le brossage 2D est une technique présente dans la plupart des systèmes de visualisation. Elle a été formalisée en particulier dans XmdvTool pour brosser des données multidimensionnelles [247]. Cette technique permet d'interactivement sélectionner un ensemble de données qui sera ensuite mis en valeur (ou masqué) dans différentes vues (linking). On peut également l'utiliser pour associer (ou retirer) une étiquette à un ensemble de données, à l'aide d'opérateurs logiques (et/ou), dans le cadre de la construction d'un clustering [74]. La brosse peut être manipulée directement, en la déplaçant sur la visualisation, ou indirectement, en modifiant ses frontières à l'aide d'un curseur de défilement (slider). Le brossage peut s'effectuer dans l'espace de représentation, c'est-à-dire en sélectionnant par manipulation directe des points sur la visualisation [17], ou bien dans l'espace des données en définissant les frontières d'un hypercube par sélection de différents intervalles de valeurs sur les dimensions [153].

Le brossage dans l'espace des données a été étendu au brossage de structures [89] basé sur un dendrogramme issu d'un clustering hiérarchique des données. Cette technique permet de brosser le dendrogramme, représenté en triangle, à une certaine hauteur et largeur, afin de définir le niveau de détails ainsi que le sous arbre à mettre en valeur. Les données brossées peuvent ensuite être mises en évidence sur des coordonnées parallèles ou des treemaps par une coloration de leurs clusters ou de leurs proximités. Ce concept est également utilisé par brossage de "collines" sur une représentation des données en profil topologique en fonction de la densité [161].

A l'inverse de ces approches qui reposent sur le brossage des structures sous-jacentes aux données à partir de leur représentation, le *brossage par proximité*, initialement introduit pour compléter un brossage 2D sur un nuage de points quelconque [160], repose sur le brossage directement dans l'espace des données d'individus similaires à un individu ou à un groupe d'individus de référence. Le brossage dans l'espace des données peut être vu comme une requête visuelle pour extraire un certain nombre d'individus pertinents. Contrairement aux approches existantes qui utilisent un hypercube sur les dimensions pour définir les bornes de la brosse, le *brossage par proximité* considère un espace des données défini par une mesure de similarité dans lequel il n'y a plus de dimensions mais uniquement des similarités entre individus. Cette approche se justifie en particulier pour des données de grande dimension pour lesquelles la définition d'un hypercube serait fastidieuse.

La lentille avec son opérateur de sélection est une technique de *brossage par proximité*. Elle permet de sélectionner des individus proches entre eux relativement à une référence dans l'espace des données. On peut donc l'utiliser pour interactivement lier la projection avec une vue des données brutes (linking). L'opérateur de la lentille permet de filtrer la visualisation de ces données brutes selon les individus sélectionnés.

Mais cette lentille permet également de suggérer des individus "brossables" directement sur la projections, c'est-à-dire des individus proches entre eux dans l'espace des données. Comme la lentille crée une zone 2D dont le périmètre est nettoyé de ses faux voisinages. La lentille permet de résoudre les problèmes de faux voisinages du brossage 2D. En effet avec un brossage 2D restreint aux points proposés par le *brossage par proximité* de la lentille, c'est-à-dire les individus dans le voisinage d'origine de la référence, il n'y a plus de problèmes de cohérence dans les groupes d'individus obtenus par brossage 2D directement sur la projection.

Nous considérons deux approches de brossage 2D :

Brossage couplé (Figure 6.17) : la brosse 2D est centrée sur le point de référence de la lentille et brosse tous les points de la lentille qui traversent le périmètre du cercle. Cette brosse "suit" interactivement la sélection de la référence et permet ainsi de brosser dans l'espace des données tout en naviguant avec la lentille sur la projection, afin de garder des traces de cette exploration.

Brossage découplé (Figure 6.18) : la brosse 2D est indépendante et se déplace avec le curseur de la souris. Cette brosse permet de choisir librement quels points brosser en fonction de la coloration des proximités d'origine relative à la référence courante.

La brosse permettant de sélectionner des points par peignage, c'est-à-dire par manipulation directe sur la projection, est représentée par un cercle de rayon γ et centré soit sur la référence, soit sur le curseur de la souris. Nous proposons d'utiliser cette technique de brossage 2D comme un *mode d'interaction* afin d'aider l'utilisateur à "extraire" sur demande des structures sous jacentes aux données dévoilées par lors de son analyse visuelle de la projection. L'activation de ce mode, que nous appellerons *mode de brossage*, fige ProxiViz et permet de brosser les points sélectionnés. Les points brossés sont ajoutés à un cluster courant sélectionnable par l'utilisateur par le biais par exemple d'une légende représentant le clustering. Des opérateurs logiques permettent ensuite de définir si un point est retiré ou ajouté au cluster courant, selon qu'il ait été déjà brossé auparavant et qu'il est à nouveau brossé.

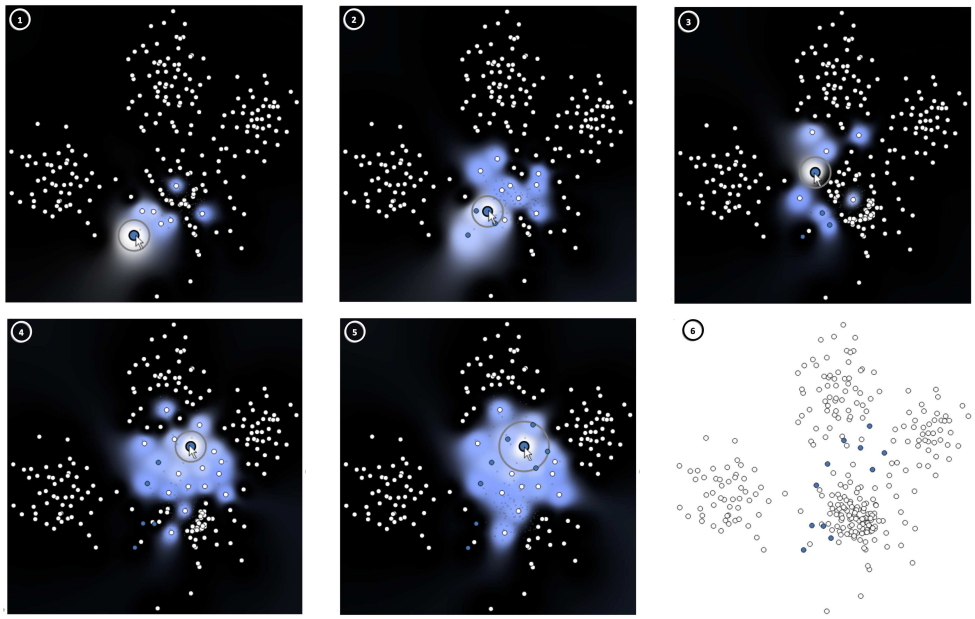


FIGURE 6.17 – Brossage 2D *couplé* à la lentille, par manipulation directe sur la projection. Les points dans le voisinage 2D et dans le voisinage d'origine sont ajoutés automatiquement au cluster courant (étiquette bleu) au cours de la navigation avec la lentille (1-5). La modification du diamètre γ de la brosse permet de sélectionner des points sans changer de référence (5). Le brossage 2D s'effectue comme l'exploration, indépendamment des faux voisinages présents sur la projection (6). Nous considérons ici un opérateur de sélection *géométrique* et un opérateur de filtrage *géométrique dépendant*.

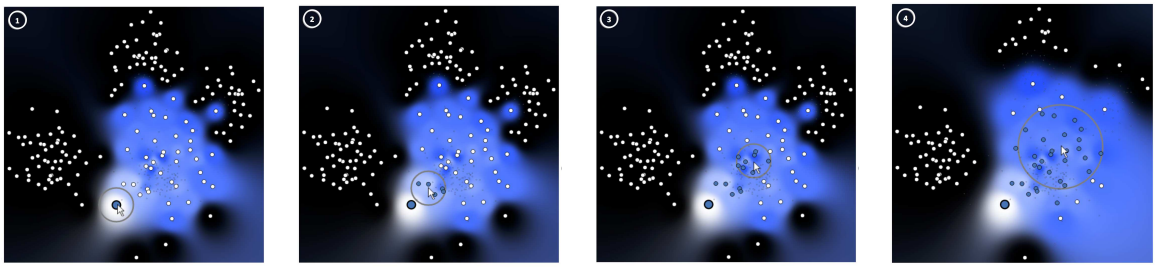


FIGURE 6.18 – Brossage 2D *découplé* de la lentille, par manipulation directe sur la projection. Les points dans le voisinage 2D et dans le voisinage d'origine sont ajoutés automatiquement au cluster courant (étiquette bleu) au cours du peignage avec la brosse (1-3). La lentille est utilisée comme un support pour localement rendre "brossable" la projection. La modification du diamètre γ de la brosse permet de sélectionner plus de points (4). Nous considérons ici un opérateur de sélection *agrégatif* et un opérateur de filtrage *géométrique dépendant*.

Interaction avec la lentille

Nous venons d'aborder la résolution des problèmes liés à l'interaction de navigation avec Proxi-Viz et du brossage 2D sur la projection, par le biais d'une lentille nettoyant localement la projection de ses faux voisinages. Cette lentille permet également de mettre en évidence sur la projection la coloration des proximités relatives à un individu de référence. Nous résumons rapidement, ci-dessous, les différents éléments de concept que nous avons introduit jusqu'ici, avant de discuter les enjeux d'interaction avec les opérateurs de la lentille.

Pour définir cette lentille, nous avons introduit un opérateur de sélection permettant de définir un voisinage de la référence dans l'espace des données et un opérateur de filtrage définissant une zone 2D sur la projection. Nous avons proposé trois exemples d'opérateurs de sélection (*géométrique, géodésique et agrégatif*) et nous avons distingué les opérateurs de filtrage *dépendant*, des opérateurs *indépendant*, selon que la zone 2D de filtrage dépende du voisinage d'origine ou non. Ces opérateurs permettent ensuite de définir une taxonomie des points par rapport à la référence.

Pour représenter de cette taxonomie, nous avons considéré différents encodages graphiques représentant le "filtrage" des faux voisinages. Pour satisfaire au critère d'équilibre entre la déformation de la projection et la mise en évidence de la coloration des proximités d'origine relatives à la référence, nous avons retenu une minimisation de la taille des faux voisinages afin de réduire les phénomènes d'occlusions, couplée à un opérateur de filtrage *géométrique dépendant* qui définit le voisinage comme l'union des cercles de rayon β centrés sur chaque point correspondant à un individu dans le voisinage de la référence dans l'espace des données. Cet opérateur de filtrage permet de minimiser la déformation de la projection tout en mettant clairement en évidence la coloration des proximités d'origine. Il est à noter que seuls les individus dans le voisinage de la référence dans l'espace des données participent à l'interpolation de la coloration de ces proximités.

Nous avons ensuite introduit différents modes d'interaction :

Mode de navigation permettant d'empêcher la sélection de faux voisinages comme référence, afin de préserver le continuum de la navigation dans l'espace des données.

Mode de sélection permettant d'inhiber temporairement la sélection automatique de la référence afin d'explorer une déchirure sur la projection. L'enveloppe convexe des points est utilisée comme opérateur de filtrage afin de mettre clairement en évidence les points sélectionnables.

Mode de brossage permettant de brosser sur la projection les points correspondant à des individus dans le voisinage d'origine de la lentille, à l'aide d'une brosse circulaire de diamètre γ étant couplée ou découplée de la référence.

Nous discutons désormais le paramétrage des opérateurs de sélection et de filtrage au regard des tâches d'analyse visuelle des projections (cf. section 2.4). L'opérateur de sélection est paramétrée par une valeur seuil α et l'opérateur de filtrage pas une valeur seuil β. Nous choisissons de laisser l'utilisateur configurer manuellement α, afin qu'il puisse appréhender la densité dans l'espace des données en adaptant manuellement l'étendue du voisinage au cours de sa navigation avec la lentille. Ce paramétrage manuel permet également de juger quantitativement les écarts de proximité là où les variations de brillance dans l'échelle de couleur ne sont pas faciles à distinguer. De la même manière, nous choisissons de laisser l'utilisateur configurer manuellement β, afin qu'il puisse contrôler manuellement l'équilibre entre la déformation de la projection et la mise en évidence de la coloration. Ce paramètre β doit être réglable manuellement par l'utilisateur car il touche directement au confort d'utilisation de la lentille.

Ces deux paramètres α et β sont contrôlable à l'aide d'un curseur de défilement (slider) et peuvent également être modifiés grâce à la molette de la souris et d'une touche raccourci, afin de ne pas quitter la projection lors de l'exploration. La configuration manuelle de ces paramètres rend l'utilisation plus souple. En revanche, il est indéniable que cela implique un certain temps d'apprentissage de la part de l'utilisateur pour maitriser l'utilisation de la lentille.

On distingue ensuite deux stratégies d'interaction avec la lentille et ses opérateurs :

Exploration : L'utilisateur navigue avec la lentille en sélectionnant différentes références, avec une même configuration des opérateurs, afin par exemple d'explorer ou d'analyser visuellement le clustering des données. Le brossage 2D couplé à la référence permet de conserver une trace de cette exploration.

Sélection : L'utilisateur fige la référence de la lentille et module les paramètres des opérateurs, afin d'analyser les proximités locales relatives à un outlier ou les proximités délimitant les frontières d'un cluster. Le brossage 2D découplé de la référence permet d'extraire un cluster mis en évidence dans la lentille par peignage directement sur la projection.

Ces deux stratégies d'interaction sont en théorie indépendantes des opérateurs, mais nous pouvons remarquer que l'opérateur *géométrique* est plus adapté à une stratégie d'exploration que l'opérateur *agrégatif*. Et réciproquement ce dernier est plus adapté à une stratégie de sélection. En effet, l'opérateur *agrégatif* définit un cluster pour une référence et une valeur de α donnée. Si cette valeur de α ne change pas (Figure 6.16-4), alors l'ensemble des individus sélectionnés reste le même tant qu'un non-voisin n'est pas sélectionné comme référence. Cette caractéristique rend la lentille "stable", ce qui est pratique pour effectuer du brossage 2D découplé de la référence, mais en contre partie elle limite les possibilités d'exploration. A l'inverse pour l'opérateur *géométrique*, l'ensemble des individus sélectionnés change dès que la référence change. En effet, le voisinage d'origine défini par cet opérateur est très "localisé" dans l'espace des données et en fonction de la topologie sous-jacente aux données, ce voisinage d'origine change d'un individu à l'autre (Figure 6.17). Aussi cet opérateur permet de naviguer et d'explorer de manière continue l'espace des données, c'est-à-dire le long de sa topologie sous-jacente.

Ces deux opérateurs n'ont pas non plus le même potentiel au regard des tâches d'analyse visuelle. En effet, l'opérateur *géométrique* est relatif à la référence dans l'espace des données, ce qui le rend moins pratique que l'opérateur *agrégatif* pour sélectionner des clusters, mais plus efficace pour étudier localement les proximités d'origine par rapport à un outlier.

Nous proposons de combiner ces deux opérateurs afin de définir une implémentation de lentille qui permette de réaliser des tâches d'analyse visuelle locale et globale. La section suivante présente une implémentation de cette lentille permettant non seulement d'aider à l'analyse visuelle mais également d'extraire un résultat par le biais du brossage 2D, comme par exemple un cluster.

6.3 ProxiLens

Nous considérons dans cette section une implémentation du concept de lentille, comme décrit précédemment, que nous appellerons ProxiLens. Cette technique combine deux opérateurs de sélection : *géométrique* et *agrégatif*, c'est-à-dire combinant un seuillage interactif des proximités d'origine relatives à une référence et la définition interactive d'un cluster contenant la référence. Ce cluster est obtenu en fonction d'une hauteur donnée dans le dendrogramme issu d'une classification ascendante hiérarchique (CAH) des données avec un critère de liaison simple. Le choix de l'opérateur de sélection est définit par l'utilisateur à la demande, par le biais d'un mode d'interaction, afin d'adapter la lentille à la tâche d'analyse visuelle à réaliser. Nous présentons les différents éléments qui composent l'interface qui sera ensuite illustrée sur les différentes tâches d'analyse visuelle avec un jeu de données d'images.

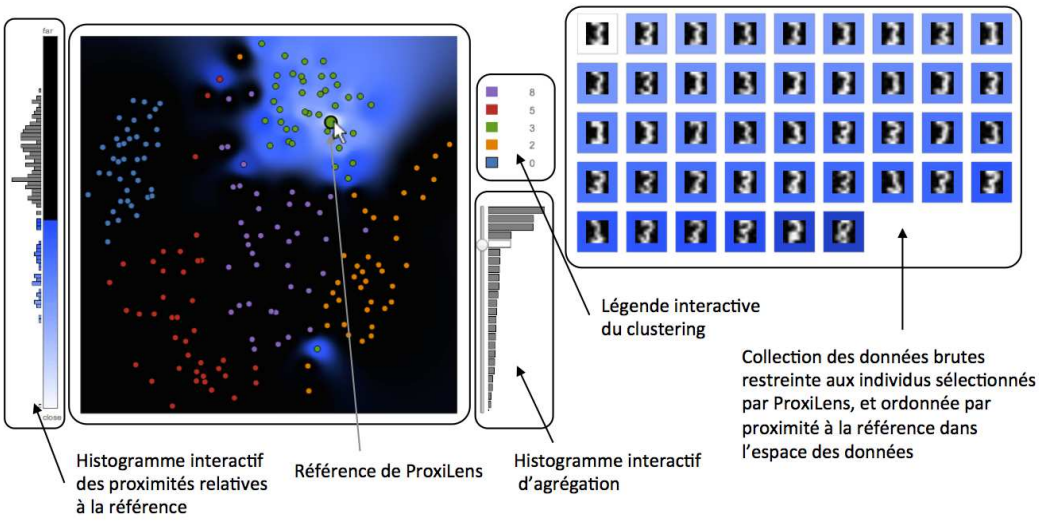


FIGURE 6.19 – Interface de ProxiLens illustrée à titre d'exemple sur une projection MDS traditionnel du jeu de données Optical Recognition of Handwritten Digits [87]. Ce jeu de données en 64 dimensions se compose d'un échantillon de 5 chiffres (0, 2, 3, 5, 8) et 40 individus choisis aléatoirement pour chaque chiffre. La lentille utilise ici l'opérateur de sélection *agrégatif*. On remarque une déchirure (en bas à droite de la projection) qui aurait été confondue avec un outlier de classe sans l'affichage des proximités d'origine. La liste des données brute (ici des images de chiffres) est limitée au voisinage de la référence (mise en évidence par une taille et un contour plus large). Cette liste est ordonnée en fonction de la proximité par rapport à la référence dans l'espace des données (les contours des images représentent cette proximité). Nous ne montrons pas sur cette figure les différents composants (sliders et sélecteurs de mode) permettant de configurer les autres paramètres comme l'opérateur de filtrage.

6.3.1 Interface

Avant d'aborder différentes tâches illustrant le fonctionnement de la technique ProxiLens, nous présentons les différents composants de l'interface que nous avons utilisé pour expérimenter cette technique. Cette interface est implémentée en `d3.js` [32] et fait partie d'un système plus large permettant de gérer l'ensemble du pipeline de réduction de dimension. Pour représenter et interagir avec les paramètres que nous avons décrit précédemment, nous considérons 4 composants d'interface en supplément de la projection (Figure 6.19) :

Histogramme interactif des proximités relatives : Ce composant est associé à l'opérateur de sélection *géométrique* de ProxiLens (Figure 6.20-1). Avec cet histogramme du nombre d'individus par *pas de proximité*, l'utilisateur peut visualiser aisément les sauts de densité dans les proximités d'origine par rapport à la référence. Ceci permet de proportionner le paramètre α_1 de l'opérateur de sélection, afin que la lentille s'étende jusqu'aux frontières d'un cluster contenant la référence ou pour détecter un éloignement de la référence par rapport aux autres individus dans l'espace des données, ce qui est caractéristique d'un outlier de données. Cet histogramme est couplé avec l'échelle de couleur qui indique visuellement la valeur de α_1, c'est-à-dire le seuil des proximités d'origine relatives à la référence.

Histogramme interactif d'agrégation : Ce composant est associé à l'opérateur de sélection *agrégatif* de ProxiLens (Figure 6.20-2). Cet histogramme indique le nombre d'individus par clusters dans le sous-arbre du dendrogramme contenant la référence. A partir de cet histogramme, l'utilisateur peut se repérer dans la hiérarchie des clusters afin de configurer le paramètre α_2 de l'opérateur de sélection. En effet, on peut visualiser les écarts de population entre les clusters de la hiérarchie, ce qui permet de différencier les jonctions du sous-arbre qui regroupent des clusters de grande taille et les jonctions qui regroupent des outliers. Cet histogramme est couplé avec un slider permettant d'indiquer et contrôler la hauteur de coupure α_2 dans le sous-arbre du dendrogramme qui contient la référence. Nous ne représentons pas le dendrogramme sous forme d'arbre afin de conserver la même abstraction qu'avec l'histogramme des proximités.

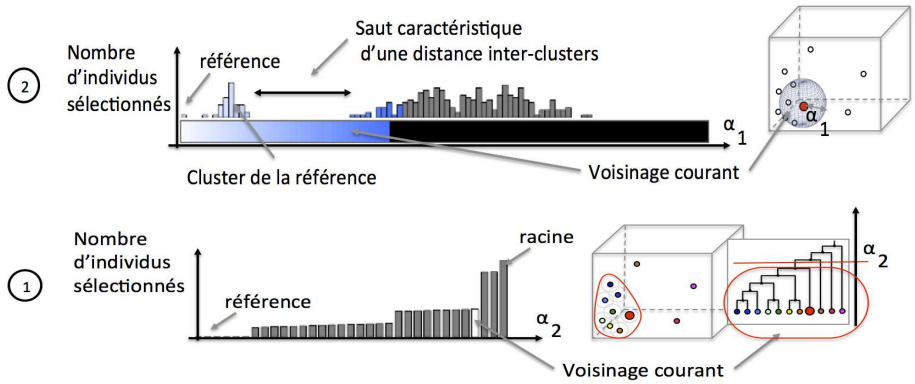


FIGURE 6.20 – Histogrammes intéractifs permettant de contrôler les paramètres α_1 de l'opérateur de sélection *géométrique* et α_2 de l'opérateur *agrégatif*.

Légende interactive du clustering : La légende permet d'indiquer les noms des classes ou des clusters. Les étiquettes de couleur sont interactives afin que l'utilisateur puisse sélectionner un cluster courant utilisé ensuite dans le *mode de brossage* de ProxiLens. L'utilisateur a le choix entre coupler ou découpler la brosse 2D de la référence selon ses besoins (par le biais d'une touche raccourci).

Représentation des données brutes : Afficher les données brutes correspondant aux individus dans la lentille permet d'utiliser la projection comme un support d'exploration des donnés. Dans le cas de données d'images, on peut remplacer les points de la projection par les images, mais cela engendrerait des problèmes d'occlusions. Aussi nous proposons de visualiser les données brutes dans une vue coordonnées à la projection et liées à l'opérateur de sélection de ProxiLens.

6.3.2 Tâches d'analyse visuelle

Dans cette sous-section, nous illustrons la technique ProxiLens sur le jeu de données CMU Face Images Dataset [87], composé d'un échantillon aléatoire de différentes personnes prises en photos (32x30 pixels, soit 960 dimensions). Ce jeu de données a été projeté avec un algorithme MDS traditionnel. Nous n'envisageons pas un scénario d'usage précis, mais nous considérons des utilisateurs souhaitant extraire des informations à partir d'une projection de leurs données. Ces utilisateurs peuvent être des analystes de données ou des non-experts.

Nous reprenons ici les différentes tâches d'analyse visuelle des projections (cf. section 2.4), selon les deux contextes d'application : exploratoire et confirmatoire. Nous montrons comment ProxiLens permet à la fois d'aider cette analyse mais également de garder des traces de l'exploration effectuée, par brossage des éléments de la structure sous-jacente aux données ayant été révélés. Dans la suite, nous appellerons *mode géométrique* et *mode agrégatif*, respectivement le mode d'interaction de ProxiLens utilisant l'opérateur de sélection *géométrique* et le mode utilisant l'opérateur de sélection *agrégatif*. Dans les figures suivantes, la liste des données brutes est affichée uniquement lorsqu'elle apporte de l'information pour effectuer la tâche d'analyse.

Analyse Exploratoire Dans ce contexte, les données ne sont pas étiquetées, c'est-à-dire qu'aucune supposition n'est faite sur le modèle à l'origine des données. On distingue trois tâches d'analyse visuelle de la projection dans un contexte exploratoire :

Détection d'outlier : ProxiLens permet de vérifier si les points 2D isolés sur la projection sont effectivement des outliers de données (Figure 6.21). En utilisant le *mode géométrique*, ProxiLens permet de juger la proximité d'un outlier candidat avec ses premiers voisins dans l'espace des données. En naviguant ensuite ces voisins, on peut juger si les relations de proximité avec cet individu sont effectivement atypiques.

Clustering visuel : ProxiLens permet de révéler les zones où le clustering 2D de la projection n'est pas fidèle au clustering dans l'espace des données (Figure 6.22). En utilisant le *mode agrégatif*, ProxiLens permet d'examiner le clustering et de visualiser les données brutes associées à chaque cluster afin de valider ces derniers. L'objectif n'est pas d'extraire un clustering des données, mais de vérifier que la projection représente fidèlement les structures sous-jacentes aux données. ProxiLens permet ensuite de brosser les différents clusters morcelés afin de construire un étiquetage permettant d'utiliser la projection comme une représentation fiable du clustering des données.

Exploration : ProxiLens permet d'utiliser la projection comme support pour explorer de manière continue l'espace des données, c'est-à-dire de proche en proche (Figure 6.23). De plus, en couplant la brosse 2D avec la référence, ProxiLens permet de brosser les points explorés afin de permettre à l'utilisateur de se repérer dans son exploration des données. La continuité de l'exploration dans l'espace des données permet de garantir une faible charge cognitive lors de l'exploration, car d'une image à l'autre il y a peu de changements. La brosse permet aussi de marquer les anomalies identifiées lors de l'exploration.

Analyse Confirmatoire Dans ce contexte, les étiquettes de chaque donnée sont connues et définissent un clustering de référence que l'on cherche à confirmer. Pour se faire, on compare le clustering formé par les étiquettes avec le clustering 2D des points de la projection. Ce contexte s'applique aussi bien à des analystes de données qu'à des non-experts. Nous utilisons pour ce contexte l'ensemble complet des individus du jeu de données initial que l'on projette avec un MDS traditionnel.

Validation des étiquettes : ProxiLens permet de mettre en évidence les structures sous-jacents aux classes en s'abstrayant des artefacts de projection. En utilisant le *mode de agrégatif*, ProxiLens permet de vérifier qu'il existe bien un cluster sous-jacent à chaque classe (Figure 6.23). En utilisant le *mode de géométrique*, ProxiLens permet de vérifier de possibles outliers de classes (Figure 6.24). La brosse 2D permet le cas échéant de corriger l'étiquette de classe d'un individu mal étiqueté en utilisant, en parallèle, la visualisation des données brutes pour vérifier ce qu'indique la mesure de similarité.

Analyse de la connectivité : ProxiLens permet d'étudier les relations de proximité entre les classes (Figure 6.25). En utilisant le *mode de géométrique*, ProxiLens permet d'étudier la proximité entre les classes en sélectionnant différentes références correspondant à des individus à la frontière du cluster sous-jacent à chacune des classes. L'exploration de la frontière commune à deux clusters permet d'en déduire la proximité entre les classes. Ceci permet de déterminer si ces classes sont séparées dans l'espace des données ou si au contraire elles se chevauchent.

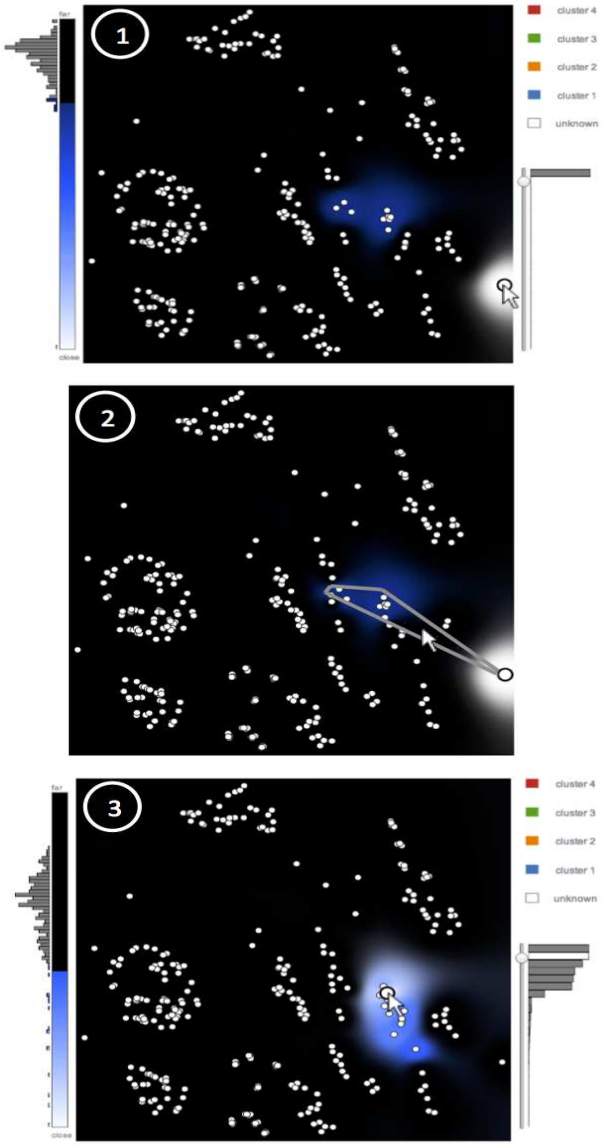


FIGURE 6.21 – Détection d'un outlier de données. On identifie visuellement un outlier 2D sur la projection puis on utilise ProxiLens pour vérifier qu'il correspond bien à un outlier de données. ProxiLens en *mode géométrique* montre que les premiers voisins de l'outlier dans l'espace des données sont très éloignés, car la couleur est très sombres (1). Ceci est confirmé par l'histogramme des proximités relatives à la référence dans l'espace des données. Afin de vérifier que la proximité de ce point à ses voisins est atypique dans l'espace des données, nous devons naviguer sur ses voisins. Pour visiter le voisinage de ce point qui correspond à une déchirure, nous utilisons le *mode de navigation* (2). L'exploration des voisins avec ProxiLens montre qu'entre eux ils sont beaucoup plus proches, car la couleur est beaucoup plus claire (3). Ceci confirme bien que notre individu de départ est un outlier de données. On remarque sur l'histogramme d'agrégation (1), que le clustering hiérarchique a effectivement identifié cet individu comme un outlier de données. En effet, on peut observer sur l'histogramme qu'entre la racine et la référence, il n'y a aucun cluster de plus d'un individu. Ceci implique que le sous arbre contenant la référence n'a qu'un seul niveau, celui connectant la référence avec la racine.

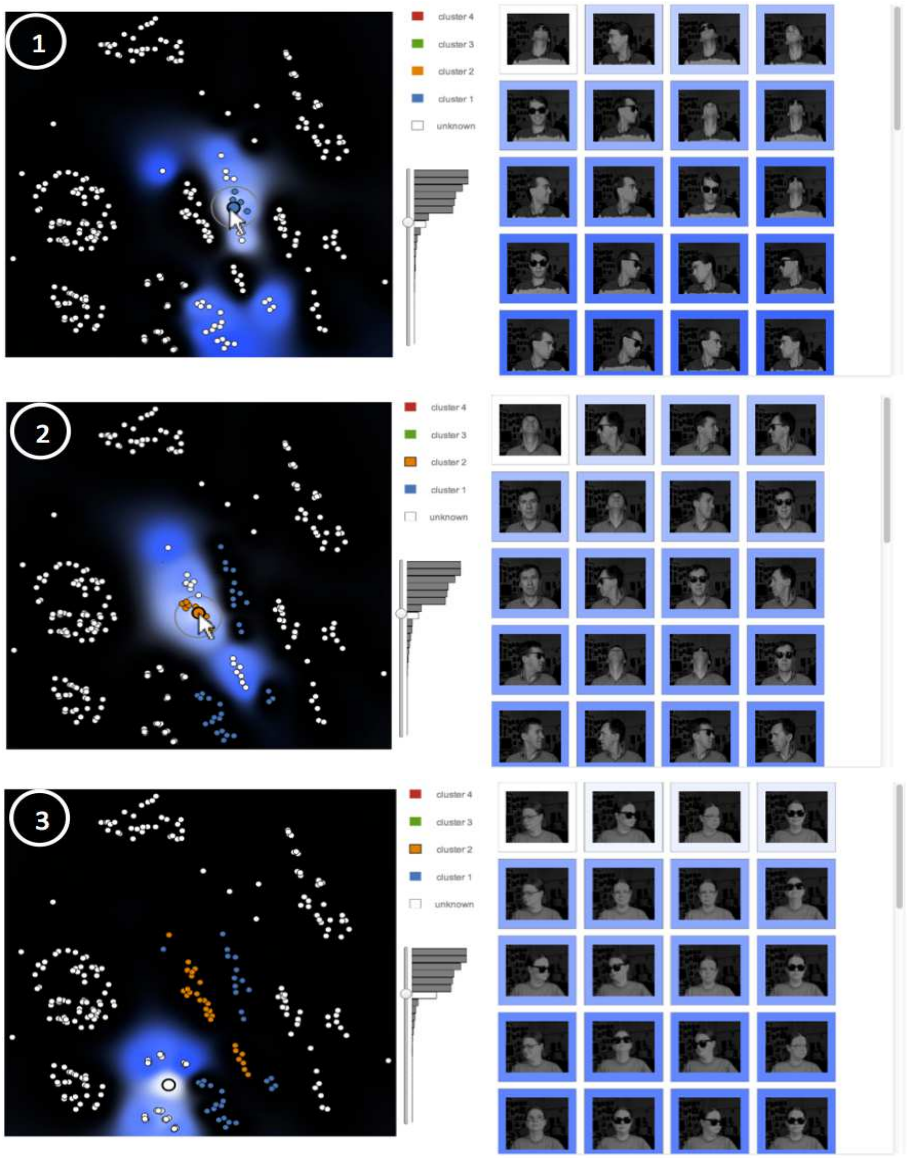


FIGURE 6.22 – Validation itérative du clustering 2D de la projection avec ProxiLens. ProxiLens en *mode agrégatif* propose un cluster englobant la référence (1). En utilisant le *mode de brossage*, on peut attribuer une étiquette (1-2) aux individus contenus dans la lentille sans problématiques d'artefacts de projection. La visualisation des données brutes permet d'aider à configurer le paramètre de sélection α_2 et de vérifier que les étiquettes correspondent à des personnes différentes dans les images. L'étiquetage itératif permet de lever les ambiguïtés sur le clustering 2D afin que la projection étiquetée mette en évidence le clustering d'origine des données.

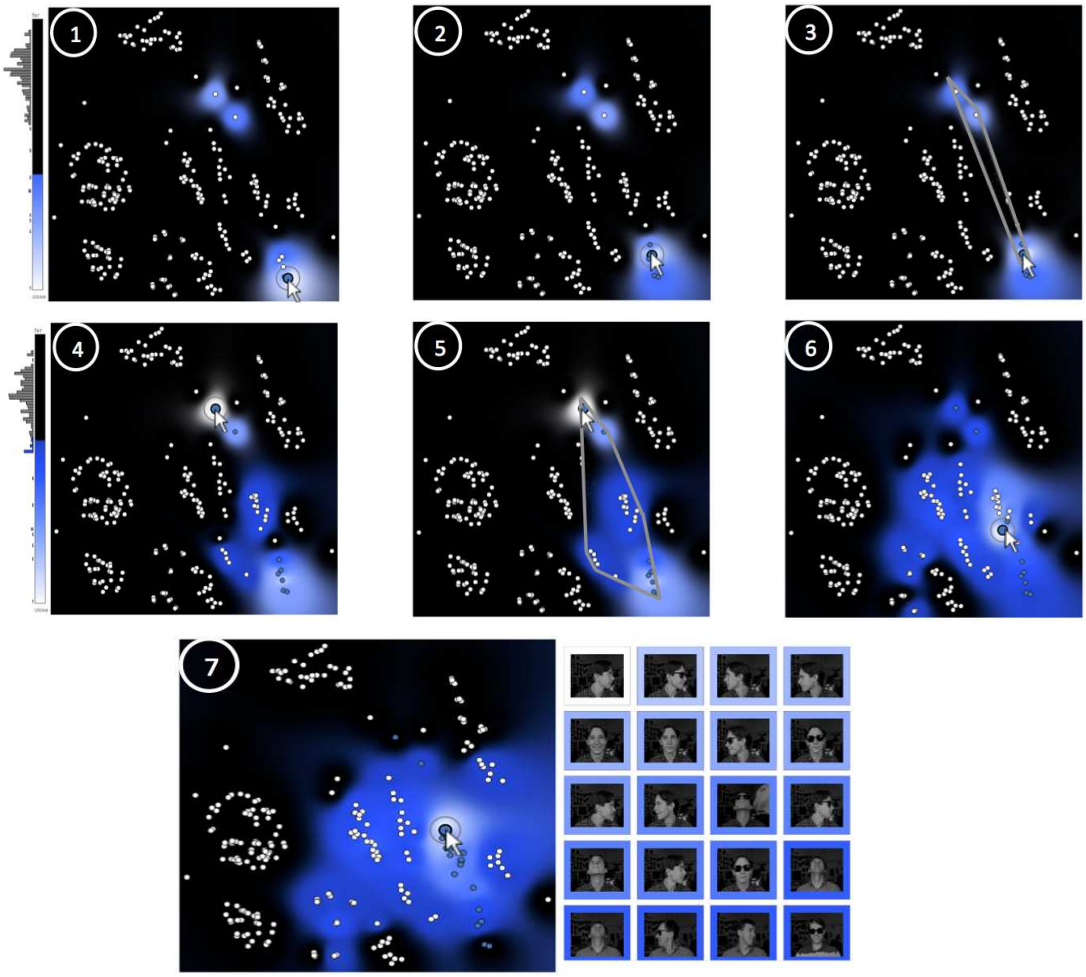


FIGURE 6.23 – Exploration des données avec ProxiLens afin de détecter d'éventuelles anomalies (non mises en valeur par la mesure de similarité). ProxiLens en *mode géométrique*, avec l'aide du *mode de navigation* (3,5) permet d'explorer de manière continue l'espace des données. Le brossage 2D couplé à la référence permet de garder une trace de l'exploration. La visualisation des données brutes permet de se repérer dans l'exploration de la collection d'image afin d'identifier les images qui correspondent aux points. L'utilisateur doit moduler la valeur du paramètre de sélection α_1 pour ne pas être bloqué par un voisinage trop restreint lors de son exploration (4), ni mettre en évidence un voisinage trop large pour lequel il devient difficile de suivre la topologie sous-jacente aux données (7).

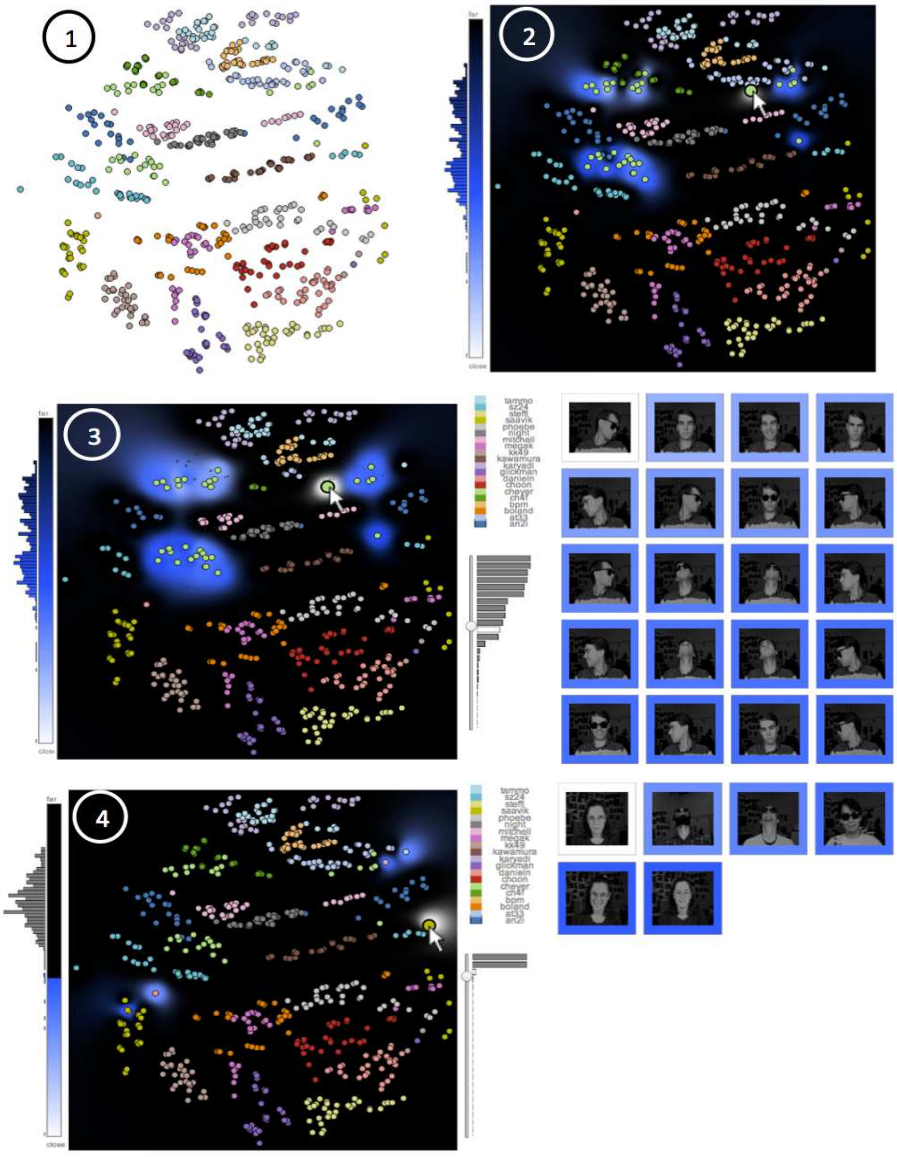


FIGURE 6.24 – Validation d'étiquettes de classe avec ProxiLens, afin de vérifier que les structures sous-jacentes aux données correspondent à un modèle a priori. ProxiLens est utilisé ici en *mode agrégatif* pour mettre en évidence le cluster sous-jacent à la classe de la référence étudiée (2). En augmentant le paramètre de filtrage β, on peut privilégier la mise en évidence de la structure par rapport à la préservation du contexte de la projection (3). ProxiLens est utilisé également en *mode géométrique* afin de vérifier de potentiels outliers de classes (4). On remarque ici que les voisins de la référence dans l'espace des données sont issus de classes différentes et correspondent également à des personnes différentes. Ceci qui tend à montrer que cette image est atypique par rapport aux autres images de cette classe, c'est-à-dire que cet individu est probablement un outlier de classe.

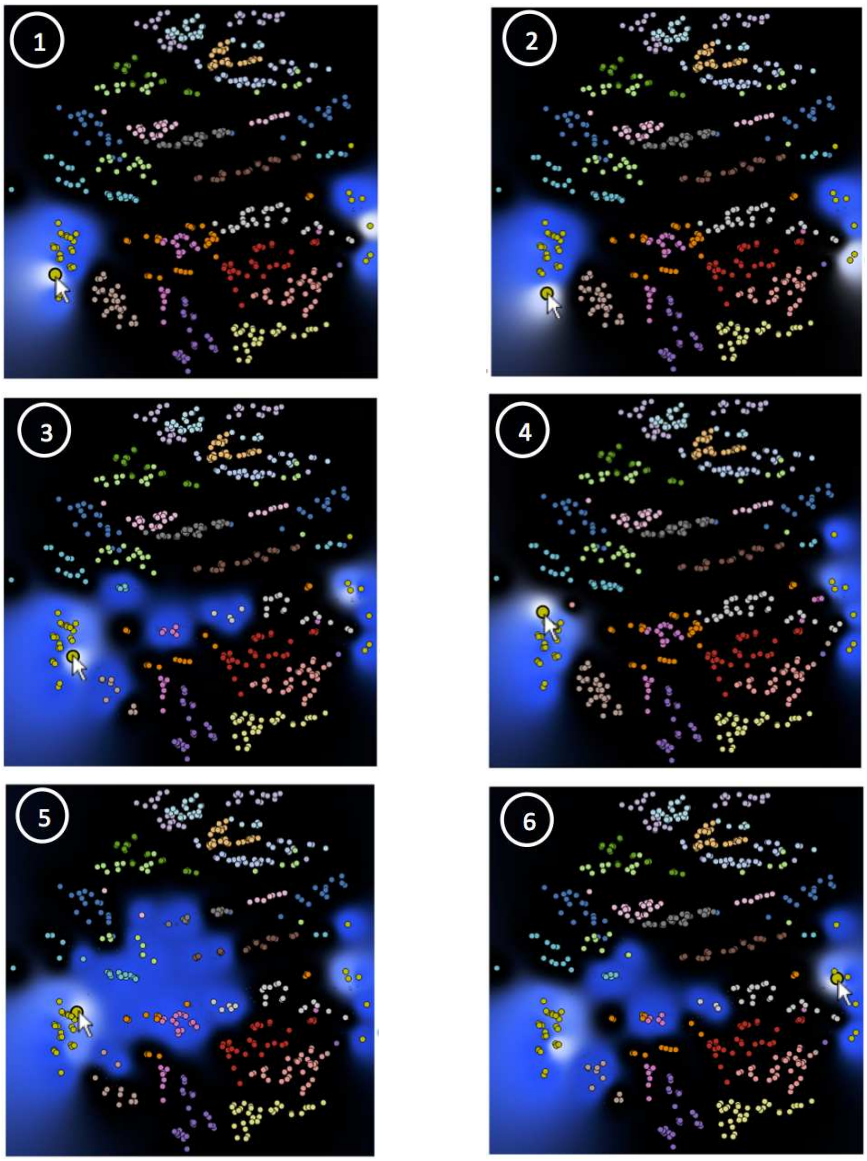


FIGURE 6.25 – Analyse de la connectivité entre classes avec ProxiLens, afin d'évaluer la proximité entre les clusters sous-jacents aux classes. ProxiLens est utilisé ici en *mode géométrique* afin de mettre en évidence des individus frontières. En explorant ensuite ces individus et en modulant éventuellement le paramètre de sélection α_1 (5), on peut mettre en évidence la proximité avec les clusters sous-jacents aux autres classes. On remarque ici qu'il n'y a pas de chevauchement entre la classes considérées et les autres classes, car il y a un saut dans l'échelle de couleur entre les points de la classe et ceux des autres classes, ce qui indique un vide dans l'espace des données. Néanmoins il faut explorer toute la topologie sous-jacente à la classe pour vérifier qu'il n'y a pas de chevauchements dans d'autres régions du cluster (1-6).

6.4 Discussion

Dans cette section, nous discutons les pistes d'amélioration de l'approche ProxiLens.

Clustering interactif La technique ProxiLens a pour vocation d'aider à réaliser les tâches d'analyse visuelle et en particulier la tâche de clustering visuel. Cette tâche consiste à utiliser ProxiLens pour vérifier le clustering 2D proposé par la projection, lequel est une approximation du clustering dans l'espace des données. Effectuer un clustering total des données n'est pas directement une finalité, mais l'opérateur de sélection *agrégatif* repose sur un dendrogramme qui fournit un clustering total pour une hauteur donnée.

Différentes approches de clustering interactif similaires existent, comme le système HCE qui repose sur la représentation d'un dendrogramme pour l'analyse de données génomiques [200, 201]. Ces approches proposent une création et évaluation interactive de clusters dans les données, afin de répondre les problèmes liés à la nature "boîte noire" des algorithmes de clustering automatique [219]. L'évaluation interactive du clustering repose sur les connaissances de l'expert pour comparer des clusters, évaluer leur stabilité, leur cohésion, ou encore suivre l'évolution d'un clustering au cours du temps avec par exemple une visualisation en ensembles parallèles [235]. Ces problèmatiques de clustering interactif sont intéressantes, mais nous choisissons d'utiliser le dendrogramme uniquement comme un moyen de suggérer des clusters dans l'espace des données car notre approche repose sur l'étude du clustering des données en vérifiant le clustering 2D de la projection, qui lui ne dépend d'aucun critère d'agrégation.

Heuristiques de paramétrage Nous avons pris le parti de laisser l'utilisateur définir manuellement les paramètres de la lentille. Ceci implique un apprentissage de la part de l'utilisateur. De plus, le paramétrage peut être une opération fastidieuse. Aussi, nous pouvons envisager des heuristiques permettant d'aider l'utilisateur dans cette démarche. Par exemple, dans un contexte confirmatoire, sur demande de l'utilisateur, le paramètre α de l'opérateur *géométrique* peut être défini comme le maximum des proximités intra-classe, afin de détecter de potentiels outliers de classe. On peut également proposer un raccourci permettant de fixer le paramètre de sélection tel que $\alpha = 1$, afin de retrouver la configuration de la technique ProxiViz. Mais ce type de paramétrages doit pouvoir être activé/désactivé sur demande de l'utilisateur afin de garantir assez de souplesse pour une utilisation exploratoire de la technique.

Liens entre mesures de similarités Nous avons proposé des opérateurs de sélection à titre d'exemple afin d'illustrer les éléments du concept. Mais on peut envisager d'autres opérateurs plus complexes, comme un opérateur qui sélectionnerait les individus voisins dans un autre espace métrique que celui de servant à la projection. Par exemple, l'espace temporel peut être utilisé pour sélectionner des individus et comparer leurs proximités 2D sur la projection avec leurs proximités dans l'espace temporel. Ceci permet de vérifier si les deux espaces sont corrélés ou non. En l'occurrence avec un opérateur de sélection dans l'espace temporel, on peut observer si les clusters visualisés sur la projection appartiennent à une même fenêtre temporelle.

Passage à l'échelle Un individu correspond à la granularité la plus faible dans l'espace des données. Pour résoudre des problèmes de passage à l'échelle, en termes de nombre de points, on peut envisager d'utiliser des prototypes. Un prototype est un individu obtenu synthétiquement à partir d'un ensemble d'autres individus, comme par exemple le barycentre de leurs vecteurs dans l'espace des données. Ce type d'individu, synthétisant des clusters, permet de réduire le nombre de points et les temps de calcul.

Qualité de la mesure de similarité Nous avons remarqué avec la pratique que l'origine des artefacts topologiques était étroitement liée à la qualité de la matrice de similarité et en particulier à la séparation des clusters dans les données en fonction de la mesure de similarité. Ce critère de séparabilité est déjà utilisé par les techniques permettant d'aider à trouver de "bonne" projection, c'est-à-dire sans trop de distorsions, mais ces techniques utilisent les clusters 2D [198]. Relier la qualité de la projection avec la séparation des clusters dans l'espace des données pose des problèmes de critères de classification. Aussi l'approche de ProxiLens est plus robuste car elle permet d'exploiter une projection indépendamment de sa qualité en termes de distorsions.

Jusqu'ici, notre approche repose sur une visualisation interactive des proximités d'origine, c'est-à-dire des informations issues directement de la matrice de similarité. Nous n'avons pas abordé les problématiques liées à cette matrice de similarité, car dans le cadre de cette thèse nous considérons une mesure de similarité fournie par un expert et supposée définir des structures sous-jacentes aux données reflétant la réalité du point de vue de cet expert. La visualisation des données brutes qui est associée à ProxiLens permet d'aider à interpréter les mesures de similarités pour caractériser ces structures. Nous avons présenté la visualisation d'une collection d'images mais on peut tout aussi bien considérer des textes, des signaux ou une heatmap de la table de données.

Cependant, il est assez difficile en pratique de définir une matrice de similarité qui reflète fidèlement la réalité. En effet, avec l'augmentation du nombre de dimensions, les distances en grande dimension souffrent d'un phénomène connu comme le "fléau de la dimension" [19] :

- Les concept de distance et de voisinage deviennent de moins en moins précis avec l'augmentation du nombre de dimensions car celle-ci fait croître très rapidement l'espace vide entre les individus. Les données deviennent d'autant plus éparses que le nombre de dimensions dépassent souvent le nombre d'individus, ce qui a pour effet de faire converger les distances entre individus vers la même valeur sur l'ensemble du jeu de données.
- L'espace dans lequel évoluent les données est complexe à interpréter. Il devient de plus en plus difficile d'énumérer et interpréter les relations entre les dimensions, ainsi que de caractériser les groupes d'individus dans la masse de dimensions.
- La pertinence des dimensions diminue, car la plupart du temps pour un groupe d'individu donné seul un petit ensemble de dimensions (*un sous-espace*), noyées dans la masse, auront des valeurs permettant de caractériser le groupe. De plus, chaque groupe d'individus peut exister dans des sous-espaces différents, ce qui rend difficile leur comparaison et séparation. Ce phénomène en particulier est connu sous le nom de pertinence ou corrélation locale des variables (*local feature relevance*). Ceci implique de chercher des sous-espaces pertinents pour chaque cluster avec des outils de sélection de variables, comme par exemple l'ACP.

Ce phénomène doit donc être pris en compte pour permettre d'utiliser l'approche de ProxiLens sur des données de très grande dimension.

6.5 Conclusion

Dans ce chapitre, nous avons d'abord introduit différentes problématiques dans la représentation et l'interaction de navigation de ProxiViz, mais également au niveau du brossage 2D sur la projection. Nous avons décrit ensuite l'espace de conception d'une lentille basée sur ProxiViz qui permet de s'affranchir de ces problématiques liées aux artefacts de faux voisinages. Finalement, nous avons proposé une implémentation de cette lentille, ProxiLens, permettant d'aider à réaliser des tâches d'analyse visuelle et nous l'avons illustré sur un jeu de données d'images.

La conceptualisation de la lentille repose sur les concepts d'opérateur de sélection dans l'espace des données et d'opérateur de filtrage dans l'espace de projection, afin de définir une taxonomie des points relativement à une référence. Nous avons proposé différents exemples d'opérateurs et défini des opérateurs de filtrage dépendant et indépendant. Nous avons ensuite considéré différents encodages graphiques de la taxonomie des points et défini un critère de choix basé sur l'équilibre entre la déformation de la projection et la mise en évidence de la coloration des proximités. Nous avons également introduit des modes d'interaction pour la navigation, la sélection de la référence et le brossage 2D, afin de résoudre les problèmes d'interaction liés aux artefacts de faux voisinages. Enfin, deux stratégies d'interaction avec la lentille et ses opérateurs ont été présentés, l'une reposant sur l'exploration de la projection et l'autre sur la variation de la taille du voisinage d'origine par rapport à la référence.

Nous privilégions un paramétrage manuel de cette lentille. Cette technique nécessite donc un certain temps d'apprentissage ce qui la rend difficile à évaluer quantitativement en termes de précision. De même, une évaluation qualitative nécessite un cas d'application avec des experts et de vraies données, car le coût d'apprentissage implique que les participants de l'étude utilisateur soient directement intéressés par les données qu'ils visualisent et qu'ils soient motivés par ce que ProxiLens peut leur apporter. Le chapitre suivant résume les différentes contributions de cette thèse et présente les travaux que nous envisageons en perspectives.

7

Conclusion et perspectives

Dans cette thèse, nous avons traité le problème général de la visualisation de données de grande dimension et plus précisément nous nous sommes intéressés à la question de l'impact des artefacts de réduction de dimension sur l'analyse visuelle des projections de données. L'approche soutenue dans cette thèse repose sur la prise en compte interactive des artefacts afin d'aider l'analyse visuelle des projections de données, c'est-à-dire améliorer la fiabilité des interprétations faites sur les données à partir de la projection.

Dans ce chapitre, nous résumons les contributions de cette thèse et nous présentons les perspectives de poursuite de ce travail.

7.1 Contributions

Après un état de l'art des techniques de visualisation de données multidimensionnelles, qui ne passent pas à l'échelle en termes de dimensionnalité, nous avons centré notre approche sur la projection par réduction de dimension afin de répondre à la problématique de la visualisation de données de grande dimension. L'état de l'art sur ces techniques de projection et sur leurs mesures de qualité associées, montre que des efforts sont fait pour aider à paramétrer ces algorithmes et obtenir une "bonne" projection. Cependant on constate un manque d'outils dans la littérature permettant d'aider à s'affranchir des problématiques d'artefacts quelle que soit la projection et sa qualité en termes de distorsions.

❶ Nous avons positionné le biais introduit par les artefacts de projection dans la chaîne de traitement par réduction de dimension (cf. chapitre 2, section 2.3) puis nous avons souligné l'absence de prise en compte des artefacts de projection dans les systèmes d'analyse basés sur les projections, ce qui pose la question de la fiabilité des analyses faites aussi bien par des experts en analyse de données que part des non-experts dans des cas d'application exploratoires ou confirmatoires. En reprenant la définition des artefacts topologiques et en insistant sur leurs propriétés de relativité et de granularité, nous avons introduit un cadre théorique avec une taxonomie des tâches de l'analyse visuelle des projections de données mettant en évidence les risques d'erreurs d'interprétation associés aux différents types d'artefacts (cf. chapitre 2, section 2.4).

Nous avons revisité la conception de la *visualisation interactive des proximités* [14] en termes d'encodage couleur et d'interaction de navigation. Nous appelons ProxiViz, cette nouvelle technique utilisant une interpolation de la couleur entre les points pour encoder les proximités d'origine relatives à un individu référence. ProxiViz étant une technique relativement simple, elle ne nécessite pas un temps d'apprentissage trop important et s'adresse aussi bien à des analystes de données qu'à des non-experts.

❷ Nous avons ensuite mis en place un cadre expérimental, avec deux expérimentations contrôlées, afin de quantifier l'impact des artefacts sur l'analyse visuelle des projections et d'évaluer les performances de ProxiViz pour l'aide à l'analyse visuelle des projections, comparé à une projection classique sans ajout d'information.

La première évaluation a mis en évidence, pour une tâche d'énumération de clusters, que la coloration interpolée est significativement plus précise que les autres variantes de coloration de ProxiViz. Les résultats ont également montré que l'ajout statique d'informations sur les distorsions n'était pas une approche pertinente pour aider l'analyse visuelle. Nous avons également noté que les projections avec des déchirures obtiennent globalement des résultats significativement meilleurs que celles avec des faux voisinages.

La seconde évaluation a montré que ProxiViz est une technique interactive efficace pour des tâches d'analyse locale à partir d'une projection comme la vérification d'outliers ou de clusters. De plus, cette technique est robuste à l'influence des artefacts de projection alors que la projection sans ajout d'information ne l'est pas pour ces tâches. En revanche, pour des tâches d'analyse globale comme l'énumération des clusters, les résultats montrent que l'influence des artefacts semble moins importante. De plus, ProxiViz n'apporte rien à la projection laquelle donne déjà un aperçu approché du clustering des données relativement satisfaisant.

❸ Enfin, nous avons étudié l'espace de conception d'une technique interactive basée sur ProxiViz et sur la métaphore de lentille, permettant de nettoyer localement la projection de ses artefacts de faux voisinages. Cette lentille permet de résoudre les problèmes d'encodage graphique et d'interaction de navigation avec ProxiViz, ainsi que de brossage 2D sur la projection liés aux artefacts de faux voisinages. La conceptualisation de la lentille repose sur les concepts d'opérateur de sélection dans l'espace des données et d'opérateur de filtrage dans l'espace de projection, afin de définir une taxonomie des points relativement à une référence. Nous avons proposé différents opérateurs, encodages graphiques et modes d'interaction, ainsi que des critères permettant de comparer et choisir parmi les possibilités de conception. Finalement, nous avons proposé une implémentation de cette lentille, ProxiLens, permettant d'aider à réaliser des tâches d'analyse visuelle et nous l'avons illustré sur un jeu de données d'images pour des contextes d'application exploratoires et confirmatoires.

7.2 Perspectives

Dans cette thèse, nous faisons l'hypothèse forte que l'on peut considérer une mesure de similarité révélant une structure sous-jacente aux données qui reflète la réalité. Nous prenons le parti d'utiliser les projections augmentées d'outils interactifs permettant de faire fi des artefacts, afin que n'importe quel utilisateur, souhaitant extraire ou utiliser la structure sous-jacente aux données, puisse exploiter une projection indépendamment de l'algorithme l'ayant généré et de sa qualité en termes de distorsions. Nous discutons ici les perspectives de poursuite de ces travaux en levant certaines des hypothèses de départ.

Vers l'interprétation de la mesure de similarité Nous avons placé en perspectives l'étude utilisateur de ProxiLens. Cette étude nécessite un cas d'application avec des experts et de vraies données, car le coût d'apprentissage pour maîtriser la technique implique que les participants de l'étude soient motivés à utiliser ProxiLens pour analyser leurs données. Pour aller dans ce sens, nous avons travaillé au développement d'un système implémentant l'ensemble du pipeline de réduction de dimension et permettant ainsi de passer des données brutes à leur visualisation et analyse en utilisant les projections de données. Ce système interface la spécification de la mesure de similarité ou de l'algorithme de projection et permet ensuite d'analyser visuellement la projection en utilisant ProxiLens. En commençant à travailler avec des experts en analyse du signal, nous avons remarqué que l'analyse de la structure sous-jacente aux données ne leur était pas suffisante et qu'ils avaient besoin de caractériser les clusters révélés par la projection pour pouvoir interpréter leurs données.

La caractérisation des clusters extraits à partir de la projection est un problème lié à l'interprétation de la mesure de similarité. Une tâche de l'analyse du clustering consiste à trouver les dimensions les plus pertinentes caractérisant un cluster dans les données, c'est-à-dire le différenciant des autres clusters. Cette tâche est également liée au problème de "fléau de la dimension" (cf. chapitre 6) et en particulier à la pertinence locale des variables (sous-espaces). Différentes techniques de visualisation existent pour aider à trouver un sous-espace pertinent [223, 224]. La projection conjointe des dimensions et des individus est une approche récente dans la littérature

de la visualisation d'information [233, 234, 272]. Cette approche permet d'analyser le clustering dans différents sous-espaces. Cependant elle n'a pas encore révélé tout son potentiel. L'utilisation de ProxiLens sur une projection conjointe des dimensions et des individus serait donc une piste intéressante, permettant de manipuler simultanément des clusters de dimensions et d'individus pour construire interactivement une mesure de similarité aidant à classifier les données.

Vers d'autres représentations Notre approche est centrée sur l'utilisation de la projection pour visualiser des données de grande dimension. Néanmoins la visualisation de la matrice de similarité sous forme de *heatmap*, que nous appellerons simplement matrice, est une autre alternative pour représenter ces données [265, 264]. Il y a des points communs entre les matrices et les projections, à commencer par le recours à un processus d'optimisation pour, dans un cas, ordonner les lignes/colonnes sur un axe 1D, et dans l'autre, positionner des points les uns par rapport aux autres dans un plan 2D. L'ordonnancement de la matrice dépend le plus souvent d'une approche de clustering des données, comme par exemple une classification ascendante hiérarchique (CAH)[15]. Aussi lorsque les clusters sont biens séparés dans les données, l'ordonnancement des lignes/colonnes de la matrice met clairement en évidence le clustering. Cependant dans le cas où les données ont une topologie sous-jacente complexe, laquelle est difficilement appréhendée par des algorithmes de clustering comme la CAH, la matrice ne révélera pas aussi bien la structure sous-jacente aux données qu'une projection de données, car celle-ci optimise la préservation de cette structure.

On retrouve dans les matrices des problèmes d'artefacts, à l'image des artefacts topologiques dans les projections. On trouve des problèmes de faux voisinages, c'est-à-dire des lignes/colonnes consécutives qui correspondent à des individus non-voisins dans l'espace des données, ainsi que des déchirures, c'est-à-dire des clusters dans l'espace des données qui sont séparés en différentes composantes non connexes sur l'axe 1D de la matrice. L'étude de l'impact de ces artefacts sur l'analyse visuelle des matrices, ainsi que la comparaison des performances d'analyse visuelle avec celles des projections de données, seraient des perspectives intéressantes à développer.

Bibliographie

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science*, pages 420–434, 2001.

[2] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. *SIGMOD Rec.*, 30(2) :37–46, 2001.

[3] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration : An implementation and evaluation. Technical report, UM Computer Science Department ; CS-TR-2763CAR-TR-584, 1992.

[4] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer London, 2011.

[5] G. Albuquerque, M. Eisemann, D. Lehmann, H. Theisel, and M. Magnor. Improving the visual analysis of high-dimensional datasets using quality measures. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 19–26, 2010.

[6] A. B. Alencar, M. C. F. de Oliveira, and F. V. Paulovich. Seeing beyond reading : A survey on visual text analytics. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 2(6) :476–492, 2012.

[7] L. Allano, M. Aupetit, and G. Sannié. Le projet eritr@c : du neutron à l'aide à la décision le projet eritr@c : du neutron à l'aide à la décision pour le contrôle de conteneurs. *WISG 2012*, 2010.

[8] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization. INFOVIS 2005*, pages 111 – 117. IEEE Computer Society, 2005.

[9] K. Andrews, C. Gutl, J. Moser, V. Sabol, and W. Lackner. Search result visualisation with xfind. In *User Interfaces to Data Intensive Systems, 2001. UIDIS 2001. Proceedings. Second International Workshop on*, pages 50–58. IEEE, 2002.

[10] N. Andrienko and G. Andrienko. *Exploratory analysis of spatial and temporal data : a systematic approach*. Springer Verlag, 2006.

[11] U. Ascher and L. Petzhodl. Computer methods for ordinary differential equations and differential-algebraic equations. *SIAM*, 1998.

[12] D. Asimov. The grand tour : a tool for viewing multidimensional data. *SIAM journal on scientific and statistical*, 6 :128–143, 1985.

[13] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sallaberry, and A. Saulnier. Gospermap : Using a gosper curve for laying out hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 19(11) :1820–1832, 2013.

[14] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7-9) :1304–1330, 2007.

[15] Z. Bar-Joseph, D. Gifford, and T. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. In *Bioinformatics*, 2001.

[16] W. Basalaj. Incremental multidimensional scaling method for database visualization, 1999.

[17] R. Becker and W. Cleveland. Brushing scatterplots. *Technometrics*, 29(2) :127–142, 1987.

[18] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *In Advances in Neural Information Processing Systems*, 14 :585–591, 2002.

[19] R. Bellman. *Adaptive Control Processes : A Guided Tour*. Princeton University Press, 1961.

[20] F. Bendix, R. Kosara, and H. Hauser. Parallel sets : A visual analysis of categorical data. In *In Proceedings of the IEEE Symposium on Information Visualization*, pages 133–140. Press, PDF/bendix2005parallelsets.pdf 2005.

[21] C. Bentley and M. Ward. Animating multidimensional scaling to visualize n-dimensional data sets. In *Proceedings IEEE Symposium on Information Visualization*, pages 72 –73, 1996.

[22] P. Berkhin. A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, and M. Teboulle, editors, *Grouping Multidimensional Data*, pages 25–71. Springer Berlin Heidelberg, 2006.

[23] E. Bertini, A. Tatu, and D. A. Keim. Quality metrics in high-dimensional data visualization : An overview and systematization. *IEEE Symposium on Information Visualization (InfoVis)*, 17(12) :2203–2212, 2011.

[24] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses : The see-through interface. In *ACM SIGGRAPH*, pages 137–145, 1993.

[25] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[26] G. Biswas, A. Jain, and R. Dubes. Evaluation of projection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(6) :701–708, 1981.

[27] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic pointing : Improving target acquisition with control-display ratio adaptation. In *Proceedings of the 22nd international conference on Human factors in computing systems (CHI 2004)*, pages 519–526, 2004.

[28] I. Borg and P. Groenen. *Modern Multidimensional Scaling : Theory and Applications*. Springer, 2005.

[29] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen. Glyph-based visualization : Foundations, design guidelines, techniques and applications. In *Eurographics State of the Art Reports*, pages 39–63, 2013.

[30] R. Borgo, K. Proctor, M. Chen, H. Janicke, T. Murray, and I. Thornton. Evaluating the impact of task demands and block resolution on the effectiveness of pixel-based visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6) :963–972, 2010.

[31] D. Borland and R. Taylor II. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, pages 14–17, 2007.

[32] M. Bostock, V. Ogievetsky, and J. Heer. D3 : Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[33] K. Boyack, B. Wylie, and G. Davidson. Domain visualization using vxinsight® for science and technology management. *Journal of the American Society for Information Science and Technology*, 53(9) :764–774, 2002.

[34] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *Proceedings of the 14th international conference on Graph drawing*, pages 42–53, 2007.

[35] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof : identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 93–104, 2000.

[36] C. Brewer. Color use guidelines for data representation. *Proc. Section on Statistical Graphics*, pages 55–60, 1999.

[37] D. Brodbeck, M. Chalmers, A. Lunzer, and P. Cotture. Domesticating bead : adapting an information visualization system to a financial institution. In *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, 1997.

[38] J. Broekens, T. Cocx, and W. Kosters. Object-centered interactive multi-dimensional scaling : Ask the expert. *Proceedings of the 18th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC)*, pages 59–66, 2006.

[39] E. T. Brown, J. Liu, R. Chang, and C. E. Brodley. Dis-function : Learning distance functions interactively. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, 2012.

[40] P. Bruneau and B. Otjacques. A proposition of interactive visual clustering system. *Proceedings of the EuroVis Workshop on Visual Analytics using Multidimensional Projections*, 2013.

[41] A. Buja, D. Swayne, M. Littman, N. Dean, and H. Hofmann. XGvis : Interactive data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, pages 1061–8600, 2001.

[42] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2) :444–472, 2008.

[43] C. J. C. Burges. Dimension reduction : A guided tour. *Foundations and Trends in Machine Learning*, 2(4), 2010.

[44] N. Cao, D. Gotz, J. Sun, and H. Qu. Dicon : Interactive visual analysis of multidimensional clusters. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12) :2581 –2590, 2011.

[45] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in information visualization : using vision to think*. Morgan Kaufmann, 1999.

[46] M. S. T. Carpendale and C. Montagnese. A framework for unifying presentation space. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 61–70, 2001.

[47] M. Chalmers. Using a landscape metaphor to represent a corpus of documents. *Spatial Information Theory A Theoretical Basis for GIS*, pages 377–390, 1993.

[48] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Proceedings of the 7th Conference on Visualization '96*, 1996.

[49] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485) :209–219, 2009.

[50] F. Chevalier, J. P. Domenger, J. Benois-Pineau, and M. Delest. Retrieval of objects in video by similarity based on graph matching. *Pattern Recogn. Lett.*, 28(8) :939–949, 2007.

[51] E. H.-h. Chi and J. Riedl. An operator interaction framework for visualization systems. In *Proceedings of the 1998 IEEE Symposium on Information Visualization*, pages 63–70, 1998.

[52] J. Chuang, D. Ramage, C. D. Manning, and J. Heer. Interpretation and trust : Designing model-driven visualizations for text analysis. In *ACM Human Factors in Computing Systems (CHI)*, pages 443–452, 2012.

[53] J. H. Claessen and J. J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12) :2310–2316, 2011.

[54] W. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[55] W. C. Cleveland and M. E. McGill. *Dynamic Graphics for Statistics*. CRC Press, Inc., 1988.

[56] D. Cook, A. Buja, J. Cabrera, and C. Hurley. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 4 :155–172, 1995.

[57] D. Cook and D. Swayne. *Interactive and Dynamic Graphics for Data Analysis*. Springer, 2007.

[58] C. D. Correa, Y.-H. Chan, and K.-L. Ma. A framework for uncertainty-aware visual analytics. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 51–58, 2009.

[59] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1) :21–27, 2006.

[60] A. Cuadros, F. Paulovich, R. Minhgim, and G. Telles. Point placement by phylogenetic trees and its application for visual analysis of document collections. *IEEE Symposium Visual Analytics Science and Technology 2007 (VAST 2007)*, pages 99–106, 2007.

[61] Culturevis. culturevis.com.

[62] G. Davidson, B. Hendrickson, D. Johnson, C. Meyers, and B. Wylie. Knowledge mining with VxInsight : Discovery through interaction. *Journal of Intelligent Information Systems*, 11(3) :259–285, 1998.

[63] G. Davidson, B. Wylie, and K. Boyack. Cluster stability and the use of noise in interpretation of clustering. In *Proc. IEEE Information Visualization*, pages 23–30, 2001.

[64] J. de Leeuw and P. Mair. Multidimensional scaling using majorization : Smacof. *R. Journ. Statistical Software*, 31(3) :1–30, 2009.

[65] V. de Silva and J. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford, 2004.

[66] P. Demartines and J. Hérault. Curvilinear component analysis : A self-organizing neural network for nonlinear mapping of data sets. *Neural Networks, IEEE Transactions on*, 8(1) :148–154, 2002.

[67] I. Dhillon, D. Modha, and W. Spangler. Class visualization of high-dimensional data with applications. *Computational statistics & data analysis*, 41(1) :59–90, 2002.

[68] M. Dork, S. Carpendale, and C. Williamson. Visualizing explicit and implicit relations of complex information spaces. *Information Visualization*, 11(1) :5–21, 2012.

[69] D. Dorling. *The Visualization of Social Spatial Structure*. Wiley, 2012.

[70] S. dos Santos and K. Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers and Graphics*, 28(3) :311 – 325, 2004.

[71] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4) :551–559, 1983.

[72] D. M. Eler, M. Y. Nakazaki, F. V. Paulovich, D. P. Santos, G. F. Andery, M. C. F. Oliveira, J. Batista Neto, and R. Minghim. Visual analysis of image collections. *Vis. Comput.*, 25(10) :923–937, 2009.

[73] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6) :1216 –1223, 2007.

[74] N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the dice : Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14 :1539 – 1148, 2008.

[75] N. Elmqvist and J. Fekete. Hierarchical aggregation for information visualization : Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3) :439–454, 2010.

[76] D. Engel, L. Hüttenberger, and B. Hamann. A survey of dimension reduction methods for high-dimensional data analysis and visualization. In C. Garth, A. Middel, and H. Hagen, editors, *VLUDS*, volume 27 of *OASICS*, pages 135–149. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2011.

[77] M. Ester, H. peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, pages 226–231, 1996.

[78] S. Fabrikant, D. Monteilo, and D. Mark. The distance-similarity metaphor in region-display spatializations. *Computer Graphics and Applications, IEEE*, 26(4) :34–44, 2006.

[79] S. I. Fabrikant. *Spatial Metaphors for Browsing Large Data Archives*. PhD thesis, University of Colorado, 2000.

[80] C. Faloutsos and K.-I. Lin. Fastmap : A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *In Proc. ACM SIGMOD*, pages 163–174, 1995.

[81] J. Fekete and C. Plaisant. Excentric labeling : Dynamic neighborhood labeling for data visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems : the CHI is the limit*, pages 512–519. ACM, 1999.

[82] J.-D. Fekete. Software and Hardware Infrastructures for Visual Analytics. *Computer*, 2013.

[83] S. J. Fernstad, J. Shaw, and J. Johansson. Quality-based guidance for exploratory dimensionality reduction. *Information Visualization*, 12(1) :44–64, 2013.

[84] M. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining : A survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3) :378–394, 2003.

[85] J. Foley, A. van Dam, S. Feiner, and J. Hughes. Computer graphics : principes and practice. *Addison-Wesley*, 1990.

[86] S. Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2 :153–174, 1987.

[87] A. Frank and A. Asuncion. UCI Machine Learning Repository. Irvine, CA : University of California, School of Information and Computer Sciences, 2010.

[88] J. Friedman and J. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23 :881–890, 1974.

[89] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Structure-based brushes : A mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(2) :150–159, 2000.

[90] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6) :1670–1690, 2009.

[91] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press Professional, 1990.

[92] G. Furnas. Generalized fisheye views. *ACM SIGCHI Bulletin*, 17(4) :16–23, 1986.

[93] E. Gansner, Y. Hu, S. Kobourov, and C. Volinsky. Putting recommendations on the map : visualizing clusters and relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 345–348. ACM, 2009.

[94] A. Gorban, B. Kegl, D. Wunsch, and A. Zinovyev. *Principal Manifolds for Data Visualization and Dimension Reduction*, volume Vol.58. Lecture Notes in Computational Science and Engineering,, 2008.

[95] A. A. Goshtasby. *Image Registration - Principles, Tools and Methods.* Advances in Computer Vision and Pattern Recognition. Springer, 2012.

[96] M. Granitzer, W. Kienreich, V. Sabol, K. Andrews, and W. Klieber. Evaluating a system for interactive exploration of large, hierarchically structured document repositories. In *IEEE Symposium on Information Visualization. INFOVIS 2004*, pages 127–134. IEEE, 2005.

[97] C. H. The use of faces to represent points in k-dimensional space graphically. *Journal American Statistical Association*, 68 :361–368, 1973.

[98] L. H. Inspect, a program system to visualize and interpret chemical data. Technical Report 22, Chemomet. Intell. Lab. Syst., 1994.

[99] R. B. Haber and D. A. Mcnabb. Visualization Idioms : A Conceptual Model for Scientific Visualization Systems. In *Visualization in Scientific Computing*. IEEE Computer Society Press, 1990.

[100] S. Haroz and D. Whitney. How capacity limits of attention influence information visualization effectiveness. *IEEE Trans. Vis. Comput. Graph.*, 18(12) :2402–2410, 2012.

[101] W. Harvey and Y. Wang. Topological landscape ensembles for visualization of scalar-valued functions. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'10, pages 993–1002. Eurographics Association, 2010.

[102] J. Heer and D. Boyd. Vizster : Visualizing online social networks. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, 2005.

[103] J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. In *Eurographics 2013 - State of the Art Reports*, 2012.

[104] G. Hinton and S. Roweis. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems*, pages 833–840, 2002.

[105] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2) :85–126, 2004.

[106] P. Hoffman, G. Grinstein, and D. Pinkney. Dimensional anchors : A graphic primitive for multidimensional multivariate information visualizations. In *In Proc of the NPIV 99*, pages 9–16, 1999.

[107] P. E. Hoffman. *Table Visualizations : A Formal Model and Its Applications*. PhD thesis, University of Massachusetts Lowell, 2000.

[108] D. Holten. Hierarchical edge bundles : Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5) :741 – 748, 2006.

[109] D. Holten, P. Isenberg, J. van Wijk, and F. J.D. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. *4th IEEE Pacific Visualization Symposium*, pages 195 – 202, 2011.

[110] D. Holten and J. Van Wijk. Evaluation of cluster identification performance for different pcp variants. *IEEE Symposium on Visualization 2010*, 29(3) :793–802, 2010.

[111] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. Websom-self-organizing maps of document collections. In *Proceedings of WSOM*, volume 97, pages 4–6, 1997.

[112] M. Hubert, P. Rousseeuw, and K. Branden. Robpca : a new approach to robust principal component analysis. *Technometrics*, 2005.

[113] C. Hurter, O. Ersoy, and A. Telea. Moleview : An attribute and structure-based semantic lens for large element-based plots. In *IEEE Transactions on Visualization and Computer Graphics 17*, pages 2600–2609, 2011.

[114] C. Hurter, B. Tissoires, and S. Conversy. Fromdady : Spreading aircraft trajectories across views to support iterative queries. *IEEE Transactions on Visualization and Computer Graphics*, 15(6) :1017–1024, 2009.

[115] I. Icke and A. Rosenberg. Automated measures for interpretable dimensionality reduction for visual classification : A user study. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 281–282, 2011.

[116] S. Ingram. *Practical Considerations for Dimensionality Reduction : User Guidance, Costly Distances, and Document Data*. PhD thesis, University of British Columbia, 2013.

[117] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. Dimstiller : Workflows for dimensional analysis and reduction. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 3–10, 2010.

[118] S. Ingram, T. Munzner, and M. Olano. Glimmer : Multilevel mds on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, pages 249–261, 2009.

[119] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2) :69–91, 1985.

[120] B. J. *Semiology of graphics*. University of Wisconsin Press, 1983.

[121] A. K. Jain. Data clustering : 50 years beyond k-means. *Pattern Recogn. Lett.*, 31(8) :651–666, 2010.

[122] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Comput. Surv.*, 31(3) :264–323, 1999.

[123] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6) :993–1000, 2009.

[124] P. Joia, F. Paulovich, D. Coimbra, J. Cuminato, and L. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12) :2563–2571, 2011.

[125] I. Jolliffe. Principal component analysis. *Springer-Verlag*, 2002.

[126] V. d. S. Joshua M. Lewis, Laurens van der Maaten. A behavioral investigation of dimensionality reduction. *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, pages 671–676, 2012.

[127] A. Kearsley, R. Tapia, and M. Trosset. The solution of the metric stress and stress problems in multidimensional scaling using newton's method. *Computational Statistics*, 13(3) :369–396, 1998.

[128] D. Keim. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1) :1–8, 2002.

[129] D. Keim, G. Andrienko, J. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics : Definition, process, and challenges. *Information Visualization*, pages 154–175, 2008.

[130] D. Keim and H. Kriegel. Visdb : a system for visualizing large databases. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, page 482. ACM, 1995.

[131] D. A. Keim. Designing pixel-oriented visualization techniques : Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1) :59–78, 2000.

[132] D. A. Keim and H.-P. Kriege. Visualization techniques for mining large databases : A comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8 :923–938, 1996.

[133] W. Kienreich, V. Sabol, M. Granitzer, F. Kappe, and K. Andrews. Infosky : A system for visual exploration of very large, hierarchically structured knowledge spaces. In *Proceedings der GI Workshopwoche, Workshop der Fachgruppe Wissensmanagement*, 2003.

[134] K. Koffka. Principle of gestalt psychology. *The Internationnal Library of Psychology*, 1947.

[135] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43 :pp.59–69, 1982.

[136] Y. Koren and L. Carmel. Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10(4) :459–470, 2004.

[137] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data : A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1) :1 :1–1 :58, 2009.

[138] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29 :1–27, 1964.

[139] S. Kullback and R. A. Leibler. On information and suciency. *The Annals of The Annals of Mathematical Statistics*, 22 :79–86, 1951.

[140] L. T. L. Bergman, B. Rogowitz. A rule-based tool for assisting colormap selection. *IEEE Visualization*, pages 118–125, 1995.

[141] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proceedings of the 1st Conference on Visualization '90*, pages 230–237, 1990.

[142] J. Lee, A. Lendasse, N. Donckers, and M. Verleysen. A robust nonlinear projection method. *Proceedings of ESANN'2000, Eighth European Symposium on Articial Neural Networks*, 2000.

[143] J. Lee and M. Verleysen. Quality assessment of dimensionality reduction : Rank-based criteria. *Neurocomputing*, 72(7-9) :1431–1443, 2009.

[144] J. A. Lee and M. Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recogn. Lett.*, 31(14) :2248–2257, 2010.

[145] J. A. Lee and M. Verleysen. Unsupervised dimensionality reduction : Overview and recent advances. *In IJCNN*, 2010.

[146] S. Lespinats and M. Aupetit. CheckViz : Sanity Check and Topological Clues for Linear and Non-Linear Mappings. *Computer Graphics Forum*, 30(1) :113–125, 2011.

[147] S. Lespinats and M. Aupetit. Classimap : a supervised mapping technique for decision support. *EuroVis 2013 Workshop on Visual Analytics using Multidimensional Projections*, 2013.

[148] S. Lespinats, B. Fertil, P. Villemain, and J. Hérault. RankVisu : Mapping from the neighborhood network. *Neurocomputing*, 72(13-15) :2964–2978, 2009.

[149] S. Lespinats, M. Verleysen, A. Giron, and G. Fertil. DD-HDS : A method for visualization and exploration of high-dimensional data. *Neural Networks, IEEE Transactions on*, 18(5) :1265–1279, 2007.

[150] H. G. Levkowitz H. Color scales for image data. *IEEE Computer Graphics and Applications*, 12(1) :72–80, 1992.

[151] L. MacDonald. Using color effectively in computer graphics. *IEEE Computer Graphics and Applications*, 19(4) :20–35, 1999.

[152] G. M. H. Mamani, F. M. Fatore, L. G. Nonato, and F. V. Paulovich. User-driven feature space transformation. *Computer Graphics Forum*, 32(3pt3) :291–299, 2013.

[153] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th conference on Visualization '95*, VIS '95, pages 271–. IEEE Computer Society, 1995.

[154] L. Martin, M. Exbrayat, G. Cleuziou, and F. Moal. Interactive and progressive constraint definition for dimensionality reduction and visualization. In F. Guillet, G. Ritschard, and D. A. Zighed, editors, *Advances in Knowledge Discovery and Management*, volume 398 of *Studies in Computational Intelligence*, pages 121–136. Springer Berlin Heidelberg, 2012.

[155] R. M. Martins, D. Coimbra, R. Minghim, and A. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Computers and Graphics*, 2014.

[156] D. Mashima, S. Kobourov, and Y. Hu. Visualizing dynamic data with maps. In *4th IEEE Pacific Visualization Symposium. PacificVis 2011*, pages 0–0. IEEE, 2011.

[157] L. C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms : A survey and experimental evaluation. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, 2002.

[158] K. Moreland. Diverging color maps for scientific visualization. *Proceedings of the 5th International Symposium on Visual Computing*, 2009.

[159] C. North and B. Shneiderman. Snap-together visualization : can users construct and operate coordinated visualizations ? *International Journal of Human-Computer Studies*, 53(5) :715–739, 2000.

[160] M. Novotny and H. Hauser. Similarity brushing for exploring multidimensional relations. *Journal of WSCG*, 14(1-3) :105–112, 2006.

[161] P. Oesterling, C. H. 0002, G. H. Weber, and G. Scheuermann. Visualizing nd point clouds as topological landscape profiles to guide local data analysis. *IEEE Trans. Vis. Comput. Graph.*, 19(3) :514–526, 2013.

[162] P. Oesterling, C. Heine, H. Janicke, and G. Scheuermann. Visual analysis of high dimensional point clouds using topological landscapes. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 113–120, 2010.

[163] P. Oesterling, G. Scheuermann, S. Teresniak, G. Heyer, S. Koch, T. Ertl, and G. Weber. Two-stage framework for a topology-based projection and visualization of classified document collections. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 91–98, 2010.

[164] C. Olson. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21 :1313–1325, 1995.

[165] J. Paiva, W. Schwartz, H. Pedrini, and R. Minghim. Semi-supervised dimensionality reduction based on partial least squares for visual analysis of high dimensional data. *Computer Graphics Forum*, 31 :1345–1354, 2012.

[166] A. Pang, C. Wittenbrink, and S. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8) :370–390, 1997.

[167] S. W. Park, L. Linsen, O. Kreylos, J. D. Owens, and B. Hamann. Discrete sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 12(2) :243–253, 2006.

[168] F. Paulovich, M. Oliveira, and R. Minghim. The projection explorer : A flexible tool for projection-based multidimensional visualization. In *Proceedings of the 10th Brazilian Symposium on Computer Graphics and Image Processing*, pages 27–36, 2007.

[169] F. V. Paulovich, D. M. Eler, J. Poco, C. P. Botha, R. Minghim, and L. G. Nonato. Piecewise laplacian-based projection for interactive data exploration and organization. In *Proceedings of the 13th Eurographics / IEEE - VGTC conference on Visualization*, pages 1091–1100, 2011.

[170] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6) :559–572, 1901.

[171] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine Learning in Python . *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

[172] R. Pinho and M. de Oliveira. Hexboard : conveying pairwise similarity in an incremental visualization space. In *Information Visualisation, 2009 13th International Conference*, pages 32–37. IEEE, 2009.

[173] R. Pinho, M. de Oliveira, et al. Incremental board : a grid-based space for visualizing dynamic data sets. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1757–1764. ACM, 2009.

[174] J. Platt. Fastmap, metricmap, and landmark mds are all nyström algorithms. *In Proc. 10th Intl. Workshop on Artificial Intelligence and Statistics*, Society for Artificial Intelligence and Statistics :261–268, 2005.

[175] E. Portes dos Santos Amorim, E. Brazil, J. Daniels, P. Joia, L. Nonato, and M. Sousa. ilamp : Exploring high-dimensional spacing through backward multidimensional projection. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 53–62, 2012.

[176] R. Rao and S. Card. The table lens : merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems : celebrating interdependence*, pages 318–322. ACM, 1994.

[177] P. Rheingans. Task-based color scale design. In *In Proceedings Applied Image and Pattern Recognition*, volume 3905, pages 35–43, 1999.

[178] J. 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*, pages 61–71, 2007.

[179] W. S. Robinson. A method for chronologically ordering archaeological deposits. *American Antiquity*, 16(293–301), 1951.

[180] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Evaluating a visualisation of image similarity as a tool for image browsing. In *Information Visualization, 1999. (Info Vis '99) Proceedings. 1999 IEEE Symposium on*, pages 36–43, 143, 1999.

[181] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing ? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 190–197, 2001.

[182] G. Ross, A. Morrison, and M. Chalmers. Coordinating views for data visualisation and algorithmic profiling. In *Coordinated and Multiple Views in Exploratory Visualization, 2004. Proceedings. Second International Conference on*, pages 3–14. IEEE, 2004.

[183] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 :2323–2326, 2000.

[184] E. A. Rundensteiner, M. O. Ward, J. Yang, and P. R. Doshi. Xmdvtool : visual interactive data exploration and trend discovery of high-dimensional data sets. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 631–631, 2002.

[185] V. Sabol. Visualisation islands : Interactive visualisation and clustering of search result interactive visualisation and clustering of search result sets. Master's thesis, Graz University of Technology, 2001.

[186] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, C-18(5) :401–409, 1969.

[187] S. San Roman, R. D. de Pinho, R. Minghim, and M. C. F. de Oliveira. A study on the role of similarity measures in visual text analytics. In *GRAPP/IVAPP*, pages 429–438, 2013.

[188] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 83–91, 1992.

[189] M. Sarkar and M. H. Brown. Graphical fisheye views. *Commun. ACM*, 37(12) :73–83, 1994.

[190] L. Saul, K. Weinberger, J. Ham, F. Sha, and D. Lee. Spectral methods for dimensionality reduction. *In Semisupervised Learning*, 2006.

[191] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5) :1299–1319, 1998.

[192] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *NIPS*, 1999.

[193] T. Schreck, M. Schüssler, K. Worm, and F. Zeilfelder. Butterfly plots for visual analysis of large point cloud data. In *Proceedings of the 16th International Conference in Central Europe on Computer Graphics (WSCG08)*, pages 33–40, 2008.

[194] T. Schreck, T. von Landesberger, and S. Bremm. Techniques for precision-based visual analysis of projected data. *Information Visualization*, 9(3) :181–193, 2010.

[195] P. Schulze-Wollgast, C. Tominski, and H. Schumann. Enhancing visual exploration by appropriate color coding. In *Proceedings of International Conference in Central Europe on Computer Graphics (WSCG2005)*, 2005.

[196] M. Sedlmair, M. Brehmer, S. Ingram, and T. Munzner. Dimensionality reduction in the wild : Gaps and guidance. Technical Report TR-2012-03, UBC Computer Science Technical Report, 2012.

[197] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE transactions on visualization and computer graphics*, 19(12) :2634–2643, 2013.

[198] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory. A taxonomy of visual cluster separation factors. *Comput. Graph. Forum*, 31(3) :1335–1344, 2012.

[199] C. Seifert, V. Saboland, and W. Kienreich. Stress maps : Analysing local phenomena in dimensionality reduction based visualisations. In *Proceedings of the 1st European Symposium on Visual Analytics Science and Technology (EuroVAST'10)*, 2010.

[200] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *IEEE Computer*, 35(7) :80–86, 2002.

[201] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2) :96–113, 2005.

[202] J. Sharko, G. Grinstein, and K. Marx. Vectorized radviz and its application to multiple cluster datasets. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6) :1444–1427, 2008.

[203] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.

[204] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.

[205] B. Shneiderman. Direct manipulation : A step beyond programming languages. *IEEE Computer*, 16(8) :57–69, 1983.

[206] B. Shneiderman. The eyes have it : A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 336–343, 1996.

[207] B. Shneiderman. Dynamic queries for visual information seeking. *Software, IEEE*, 11(6) :70–77, 2002.

[208] R. Sibson. A vector identity for the dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, pages 151–155, 1980.

[209] S. Silva, B. Sousa Santos, and J. Madeira. Using color in visualization : A survey. *Computers & Graphics*, 2010.

[210] V. D. Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *In Advances in Neural Information Processing Systems*, 15 :705–712, 2003.

[211] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. In *Proceedings of the 11th Eurographics Conference on Visualization*, pages 831–838, 2009.

[212] T. Soukup and I. Davidson. *Visual Data Mining : Techniques and Tools for Data Visualization and Mining*. John Wiley & Sons, Inc., 2002.

[213] M. Spenke and C. Beilken. Infozoom-analysing formula one racing results with an interactive data mining and visualisation tool. In *International Conference on Data Mining*, pages 455–64, 2000.

[214] J. Stasko, C. Görg, and Z. Liu. Jigsaw : supporting investigative analysis through interactive visualization. *Information visualization*, 7(12) :118–132, 2008.

[215] C. Stolte, D. Tang, and P. Hanrahan. Polaris : A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1) :52–65, 2002.

[216] D. F. Swayne, D. Cook, and A. Buja. Xgobi : Interactive dynamic data visualization in the x window system. *Journal of Computational and Graphical Statistics*, 7 :113–130, 1998.

[217] D. F. Swayne, D. T. Lang, A. Buja, and D. Cook. Ggobi : Evolving from xgobi into an extensible framework for interactive data visualization. *Comput. Stat. Data Anal.*, 43(4) :423–444, 2003.

[218] S. Tableau. Tableau home page.

[219] P. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Addison Wesley Boston., 2006.

[220] K. Tasdemir and E. Merényi. Exploiting data topology in visualization and clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, 20(4) :549–562, 2009.

[221] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnork, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 59–66. IEEE, 2009.

[222] A. Tatu, P. Bak, E. Bertini, D. Keim, and J. Schneidewind. Visual quality metrics and human perception : an initial study on 2d projections of large multidimensional data. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pages 49–56, 2010.

[223] A. Tatu, F. Maas, I. Farber, E. Bertini, T. Schreck, T. Seidl, and D. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 63–72, 2012.

[224] A. Tatu, L. Zhang, E. Bertini, T. Schreck, D. Keim, S. Bremm, and T. von Landesberger. Clustnails : Visual analysis of subspace clusters. *Tsinghua Science and Technology*, 17(4) :419–428, 2012.

[225] E. Tejada, R. Minghim, and L. G. Nonato. On improved projection techniques to support visual exploration of multi-dimensional data sets. *Information Visualization*, 2(4) :218–231, 2003.

[226] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500) :2319–2323, 2000.

[227] C. Tominski, G. Fuchs, and H. Schumann. Task-driven color coding. In *Proceedings of the 12th International Conference on Information Visualisation (IV08)*, pages 373–380, 2008.

[228] W. Torgerson. Multidimensional scaling : I. Theory and method. *Psychometrika*, 17(4) :401–419, 1952.

[229] M. Tory, D. Sprague, F. Wu, W. So, and T. Munzner. Spatialization design : Comparing points and landscapes. *IEEE Transactions on Visualization and Computer Graphics*, pages 1262–1269, 2007.

[230] M. Tory, C. Swindells, and R. Dreezer. Comparing dot and landscape spatializations for visual memory differences. *IEEE Transactions on Visualization and Computer Graphics*, 15(6) :1033–1040, 2009.

[231] B. Trumbo. Theory for coloring bivariate statistical maps. *The American Statistician*, 35(4) :220–226, 1981.

[232] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[233] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions& a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12) :2591–2599, 2011.

[234] C. Turkay, A. Lundervold, A. J. Lundervold, and H. Hauser. Representative factor generation for the interactive visual analysis of high-dimensional data. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12) :2621–2630, 2012.

[235] C. Turkay, J. Parulek, N. Reuter, and H. Hauser. Integrating cluster formation and cluster evaluation in interactive visual analysis. In *Proc. Spring Conference on Computer Graphics (SCCG 2011) – second best paper*, 2011.

[236] A. Ultsch. U-matrix : a tool to visualize clusters in high dimensional data. Technical report, University of Marburg, 2003.

[237] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 2008.

[238] L. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction : A comparative review, 2008.

[239] F. van Ham and J. J. van Wijk. Interactive visualization of small world graphs. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 199–206, 2004.

[240] J. van Wijk. The value of visualization. In *Visualization, 2005. VIS 05. IEEE*, pages 79–86, 2005.

[241] J. J. van Wijk. Unfolding the Earth : Myriahedral Projections. *Cartographic Journal, The*, 45(1) :32–42, 2008.

[242] J. Venna. *Dimensionality reduction for visual exploration of similarity structures*. Dissertations in computer and information science, Helsinki University of Technology, 2007.

[243] J. Venna and S. Kaski. Local multidimensional scaling. *Neural Networks*, 19(6-7) :889–899, 2006.

[244] J. Venna and S. Kaski. Nonlinear dimensionality reduction as information retrieval. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, pages 568–575, 2007.

[245] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Infomation retrieval perspective to nonlinear dimensionality reduction. *Journal of Machine Learning Research*, 11(1) :451–490, 2010.

[246] J. Vesanto. Som-based data visualization methods. *Intelligent Data Analysis*, 3 :111–126, 1999.

[247] M. Ward. Xmdvtool : integrating multiple methods for visualizing multivariate data. In *Visualization, 1994., Visualization '94, Proceedings., IEEE Conference on*, pages 326–333, 1994.

[248] M. Ward, G. Grinstein, and D. Keim. *Interactive Data Visualization : Foundations, Techniques, and Applications*. A K Peters Ltd, Natick, MA, USA, 2010.

[249] C. Weaver. Visualizing coordination in situ. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 165–172. IEEE, 2005.

[250] C. Weaver. Improvise : a user interface for interactive construction of highly-coordinated visualizations, 2006.

[251] C. Weaver. Multidimensional visual analysis using cross-filtered views. In *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*, pages 163–170. IEEE, 2008.

[252] C. Weaver. Conjunctive visual forms. *IEEE Transactions on Visualization and Computer Graphics*, 15(6) :929–936, 2009.

[253] G. Weber, P.-T. Bremer, and V. Pascucci. Topological landscapes : A terrain metaphor for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6) :1416–1423, 2007.

[254] K. Q. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *In AAAI*, 2006.

[255] M. Wijffelaars, R. Vliegen, J. van Wijk, and E. van der Linden. Generating color palettes using intuitive parameters. *Comp. Graph. Forum*, 27(4) :743–750, 2008.

[256] L. Wilkinson. *The grammar of graphics*. Springer Verlag, 2005.

[257] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *IEEE Symposium on Information Visualization. INFOVIS 2005*, pages 157–164. IEEE, 2005.

[258] L. Wilkinson and M. Friendly. The history of the cluster heat map. *The American Statistician*, 63(2) :179–184, 2009.

[259] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *IEEE Symposium on Information Visualization. INFOVIS 2004*, pages 57–64. IEEE, 2005.

[260] G. J. Wills. Selection : 524, 288 ways to say 'this is interesting'. *Information Visualization '96 Conference Proceedings*, pages pages 54–61, 1996.

[261] J. A. Wise. The ecological approach to text visualization. *J. Am. Soc. Inf. Sci.*, 50(13) :1224–1233, 1999.

[262] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual : Spatial analysis and interaction with information from text documents. *Proceedings of the 1995 IEEE Symposium on Information Visualization*, 1995.

[263] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. *Scientific Visualization Overviews, Methodologies, and Techniques*, pages 3–33, 1997.

[264] H.-M. Wu, Y.-J. Tien, and C.-h. Chen. Gap : A graphical environment for matrix visualization and cluster analysis. *Comput. Stat. Data Anal.*, 54(3) :767–778, 2010.

[265] H.-M. Wu, S. Tzeng, and C.-h. Chen. Matrix visualization. In *Handbook of Data Visualization*, Springer Handbooks Comp.Statistics, pages 681–708. Springer Berlin Heidelberg, 2008.

[266] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser : Bridging information visualization with automated intelligent image analysis. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 191–198, 2006.

[267] J. Yang, D. Hubball, M. Ward, E. Rundensteiner, and W. Ribarsky. Value and relation display : Interactive visual exploration of large data sets with hundreds of dimensions. *Visualization and Computer Graphics, IEEE Transactions on*, 13(3) :494 –507, 2007.

[268] J. Yang, A. Patro, S. Huang, N. Mehta, M. Ward, and E. Rundensteiner. Value and relation display for interactive exploration of high dimensional datasets. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 73 –80, 2004.

[269] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6) :1224–1231, 2007.

[270] J. S. Yi, R. Melton, J. Stasko, and J. A. Jacko. Dust & magnet : Multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4) :239–256, 2005.

[271] G. Young and A. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3 :19–22, 1938.

[272] X. Yuan, D. Ren, Z. Wang, and C. Guo. Dimension projection matrix/tree : Interactive subspace visual exploration and analysis of high dimensional data. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12) :2625–2633, 2013.

[273] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3) :1047–1054, 2008.

[274] X. Zhu and J. Lafferty. Harmonic mixtures : combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 1052–1059, 2005.

[275] J. L. Zinnes and D. MacKay. Probabilistic multidimensional scaling : Complete and incomplete data. *Psychometrika*, 48(1) :27–48, 1983.