Bundling, Simplification de Graphes par Agrégation Visuelle : Etat de l'Art et Défis

Antoine Lhuillier, Christophe Hurter

Bundling, Simplification de Graphes par Agrégation Visuelle : Etat de l'Art et Défis

Antoine LHUILLIER
LII-ENAC
31055, Toulouse, France
antoine.lhuillier@enac.fr

Christophe HURTER
LII-ENAC
31055, Toulouse, France
christophe.hurter@enac.fr

RESUME
Dans le cas d'une forte densité et d'un fort taux d'occlusion, il devient difficile de distinguer les nœuds et les liens qui composent un graphe. Plusieurs techniques de bundling permettent la simplification visuelle de graphes en agrégeant les liens pour créer des zones de fortes densités au profit d'espaces plus clairsemés et ainsi faire émerger des structures visuelles. Dans cet article nous avons rassemblé des techniques de visualisation et d'interaction pour la simplification visuelle et l'exploration de graphes. Notre but est de mieux comprendre les techniques existantes et de discuter de leurs avantages et inconvénients. Nous donnons des cas d'utilisation avérés de ces techniques et nous concluons par leurs futurs défis et les possibles évolutions dans le domaine de la visualisation de graphes.

Mots Clés
Bundling ; Simplification de graphe ; InfoVis ; Visualisation de données ; Agrégation visuelle ;

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

La visualisation de graphes est un domaine de recherche très actif avec de nombreuses publications chaque année dans les conférences majeures du domaine de l'InfoVis, SciVis, Graph Drawing et du Visual Analytics [45][4]. Parmi l'ensemble des techniques de visualisation de graphes, il existe le sous-ensemble des simplifications de graphe. Les techniques d'agrégation que nous présentons dans ce papier font partie de ce sous-ensemble. Le bundling, que l'on peut traduire en français par « agrégation visuelle », permet la simplification visuelle de graphes en regroupant plusieurs liens en tronçons principaux qui se ramifient comme des rivières vers la source. Cette agrégation forme des zones plus denses au profit de zones plus clairsemées et diminue le rapport de surface encre/fond *(ink ratio)* introduit par Tufte [46]. Le bundling permet par exemple l'émergence d'informations dans un graphe à forte densité dont la visualisation sans traitement n'affiche qu'une forme compacte (figure 1). L'agrégation de tels graphes fait apparaître des structures visuelles auparavant occultées par la forte densité de ses liens.

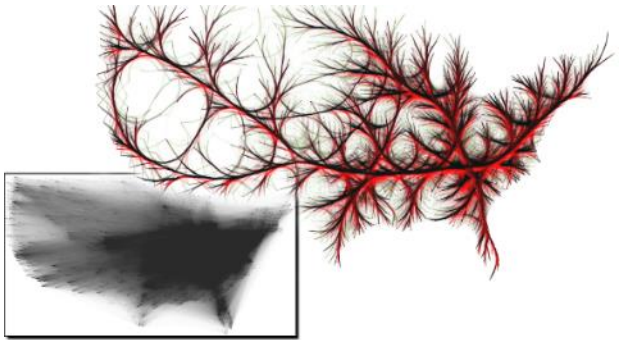


Figure 1. Graphe des migrations sur une année de populations aux Etats-Unis. Version originale en bas, agrégée en haut [24]

Aujourd'hui, plusieurs techniques de bundling existent, et le domaine est devenu plus mature grâce à de nombreux exemples d'applications. Cependant, aucun des travaux précédemment publiés n'ont, jusqu'à présent, essayé de structurer l'espace de design. Cet article est une version étendue du tutoriel sur la simplification visuelle de graphes [44]. Nous y présentons et structurons les différents algorithmes de bundling existants. Notre but est de clarifier les possibilités offertes par les outils de bundling existants et de discuter de leurs avantages et de leurs faiblesses.

L'ensemble des techniques de bundling se fondant sur d'autres techniques d'infographie, nous expliquerons dans un premier temps, les différents concepts nécessaires à la compréhension des différents algorithmes de bundling. Puis nous décrirons de manière synthétique chaque algorithme dont nous résumons les caractéristiques dans un tableau récapitulatif (table 1). Ensuite, nous discuterons des possibilités offertes par ces techniques. Pour finir, nous ouvrirons cette discussion sur les futurs défis du bundling.

DEFINITIONS

Un graphe G est composé d'un ensemble de sommets S et d'une liste d'arrêtes A les connectant (G = {S, A}). Au cours de notre analyse, nous noterons N le nombre de nœuds d'un graphe et L le nombre de liens de ce dernier.

http://dx.doi.org/10.1145/2820619.2820629

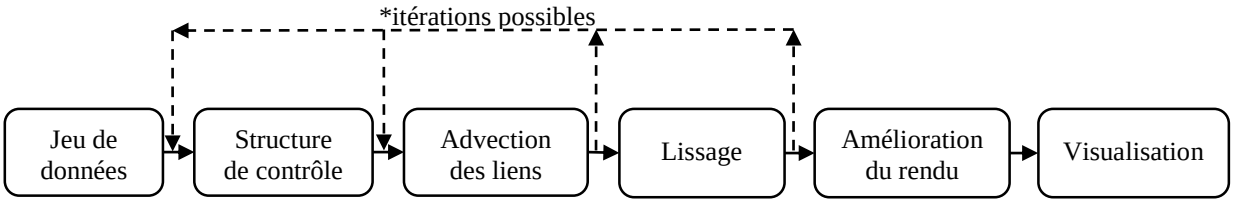


Figure 2. Pipeline Graphique du Bundling

Dans la section suivante, nous explicitons les techniques d'infographie utiles aux algorithmes de bundling : Spline, Clustering de graphes et Bump-mapping.

Spline [3]

Une cerce *(Spline)* est une courbe lisse définie en mettant bout à bout plusieurs courbes lisses définies par des points de contrôles. Il s'agit d'une fonction polynomiale définie par morceaux Les techniques de bundling utilisent principalement les B-Splines [3] qui sont la généralisation des courbes de Bézier.

Clustering de Graphes [40]

La fragmentation *(Clustering)* de graphes consiste à regrouper les nœuds et les liens d'un graphe selon des critères de compatibilité préalablement définis. Ces critères peuvent porter sur tout attribut d'un nœud ou d'un lien (poids, distance, direction, etc…). L'objectif de la fragmentation de graphes consiste à faire en sorte que la distance entre les différents sous-ensembles *(Cluster)* formés soit la plus grande possible.

Bump-Mapping

La topographie d'aspérité *(Bump Mapping)* introduite par Blinn [6] permet de modifier le rendu d'une surface. Elle ajoute du relief à une surface grâce à l'interaction de la lumière et d'une texture irrégulière. Il s'agit d'une ombre pixelaire permettant de rendre la surface bosselée en modifiant la normale de la surface sur laquelle le bump-mapping est appliqué. Un exemple de bump-mapping est visible en figure 4 avec l'effet d'ombrage.

Pipeline Graphique des Algorithmes de Bundling

Le bundling est une technique de visualisation graphique dont le pipeline graphique est variable d'une implémentation à l'autre. Néanmoins, nous proposons de généraliser ce pipeline afin de saisir le principe fondateur de tout algorithme de bundling. Ce pipeline, similaire à celui d'Infovis [17], introduit les deux notions suivantes :

Advection

Nous utiliserons la notion d'advection pour caractériser les algorithmes de bundling. Le terme d'advection désigne, dans notre cas, le transport d'une quantité de matière selon un champ vectoriel.

Structure de Contrôle

Le terme « structure de contrôle » désigne un guide servant à l'advection des liens du graphe agrégé. Il peut regrouper des règles d'agrégation selon un critère donné [21][36], un maillage [10][29] ou encore des cartes de densité [24][36].

Notre généralisation du pipeline graphique des algorithmes de bundling est montrée en figure 2 et comporte les étapes suivantes :

- la création d'une structure de contrôle,
- l'advection des liens du graphe,
- le lissage,
- l'amélioration du rendu graphique,
- la visualisation du graphe.

Les techniques de bundling se résument par la suite d'actions suivantes. L'analyse d'un jeu de données permet la construction d'une structure de contrôle qui définit les règles d'agrégation à appliquer sur le graphe. A partir de ces règles, l'advection des liens du graphe est réalisée. Ensuite, l'algorithme lisse les liens entre chaque nœud et améliore le rendu graphique. Enfin, le résultat agrégé est affiché. Selon la technique de bundling, il est possible d'itérer entre les différentes étapes (liens en pointillés de la figure 2).

Nous proposons d'utiliser ce pipeline graphique comme cadre d'analyse de l'état de l'art.

ETAT DE L'ART

Dans cette partie, nous décrivons des techniques permettant le calcul de graphes agrégés. Ces techniques opèrent toutes une déformation des liens entre les nœuds d'un graphe.

Dickerson et al. ont été les premiers à introduire le principe d'agrégation de liens dans un graphe pour en améliorer sa lisibilité. Ils utilisent un algorithme d'heuristique pour illustrer leur concept [11]. Plusieurs algorithmes ont été élaborés pour agglomérer les liens d'un graphe [38][49] ; Dwyer et al. ont utilisé des liens courbes avec un algorithme à base de force et d'anti-croisement [12]. En règle générale, peu d'implémentations du bundling modifient la position des nœuds du graphe. On peut néanmoins citer Gansner et Koren qui ont été les premiers à changer la position des nœuds dans un graphe circulaire pour optimiser l'agrégation visuelle [13]. Plus récemment Bothorel et al. [7] ont appliqué la même technique pour optimiser la visualisation des résultats de l'algorithme « apriori » [1].

Techniques de Calcul des Structures de Contrôles

Parmi l'ensemble des techniques de bundling étudiées (visibles en figure 3), nous différencions deux types d'implémentations : les techniques dites géométriques et

celles pixelaires. Cette dichotomie s'opère principalement sur l'implémentation de la structure de contrôle.

Techniques Géométriques

Ces techniques se basent sur les relations géométriques entre les nœuds du graphe.

HEB (Hierachical Edge Bundling) [22]. Cette technique se fonde sur l'analyse de données hiérarchiques. La structure hiérarchique à agréger permet de positionner des points de contrôles de B-Splines (chaque nœud est un point de contrôle). Ainsi, elle agrège visuellement les liens du graphe. Il s'agit d'une technique simple à implémenter puisqu'il suffit d'assigner chaque nœud du graphe à une B-Spline (figure 5). Sa complexité de calcul dépend du nombre de liens présents dans le graphe (i.e. en O(L)). Les paramètres sont réduits à ceux des B-Splines

FDEB (Force-Directed Edge Bundling) [21]. Très populaire, FDEB utilise la notion de compatibilité de liens pour déterminer leur pouvoir d'attraction mutuelle. Cette notion de compatibilité englobe un très grand nombre de paramètres comme par exemple la direction des liens. Très flexible et simple à implémenter, elle a le revers de disposer d'une complexité de l'ordre de $O(L^2)$ (des optimisations sont néanmoins possibles). Les paramètres de ce bundling sont définis par les fonctions de compatibilité.

GBEB (Geometry-Based Edge bundling) [10]. Cette technique utilise un maillage (méthode de fragmentation de graphe) selon la direction des liens du graphe. Ce maillage permet de définir la position des points de contrôle de B-Splines. Le calcul de ce maillage possède une complexité en O(L.log(L)) ainsi que de nombreux paramètres.

WR (Winding Road) [29]. Utilisant le même principe que GBEB, cet algorithme fonde le calcul du maillage sur un diagramme de Delaunay et Voronoï [48]. Il s'agit d'une implémentation GPU. La complexité de cette technique est similaire à celle de GBEB en O(L.log(L)). Elle possède les mêmes paramètres que GBEB.

Mingle [14]. Grâce à un prétraitement des données pour construire un graphe de proximité, le traitement des liens du graphe est parallélisé pour calculer rapidement les déplacements à opérer. L'ensemble des liens est ainsi agrégé pour former un ensemble de lignes brisées *(Polyline)*. L'implémentation de l'algorithme est complexe avec de nombreux paramètres. Cette technique est sans doute une des plus rapides, mais visuellement ne produit pas la version la plus simplifiée (figure 3). Sa complexité est de l'ordre de O(L.log(L)).

Techniques Pixelaires

Ces techniques exploitent les capacités de parallélisation des GPU modernes afin d'améliorer les performances du bundling.

SBEB (Skeleton-Based Edge Bundling) [13]. SBEB utilise un squelette de graphe qui doit être auparavant fragmenté. Ce squelette attire les liens du graphe et densifie ainsi ce dernier autour de cette structure. La complexité de cet algorithme réside dans le calcul de ce squelette. Cette technique ne nécessite que deux paramètres : un taux d'attraction ainsi qu'un taux de lissage. Sa complexité est proche de O(L) sans prendre en compte le calcul du squelette.

KDEEB (Kernel Density Estimation Edge Bundling) [24]. KDEEB calcule une carte de densité à partir des nœuds et des liens du graphe pour définir une carte de gradient pour l'advection. Ainsi, les liens du graphe sont attirés dans les zones de fortes densités. Cette technique est sans doute la plus rapide puisque sa complexité est en O(L) et ne nécessite pas de post-traitement de données. Les paramètres utilisés sont la taille de la carte de densité, le taux d'attraction et le taux de lissage.

ADEB (Attribute Driven Edge Bundling) [36]. Similaire à KDEEB, cette technique se différencie par la réutilisation des critères de compatibilité introduit par FDEB. Ainsi, ADEB intègre les critères de compatibilité de liens (pour exemple la direction d'un lien ou encore l'horodatage d'une trajectoire) pour créer différentes carte de densité. Les paramètres utilisés sont ceux de KDEEB (taille de la carte, taux d'attraction et taux de lissage) auxquels s'ajoutent les critères de compatibilité. Sa complexité reste en O(L).

Advection des Liens du Graphe

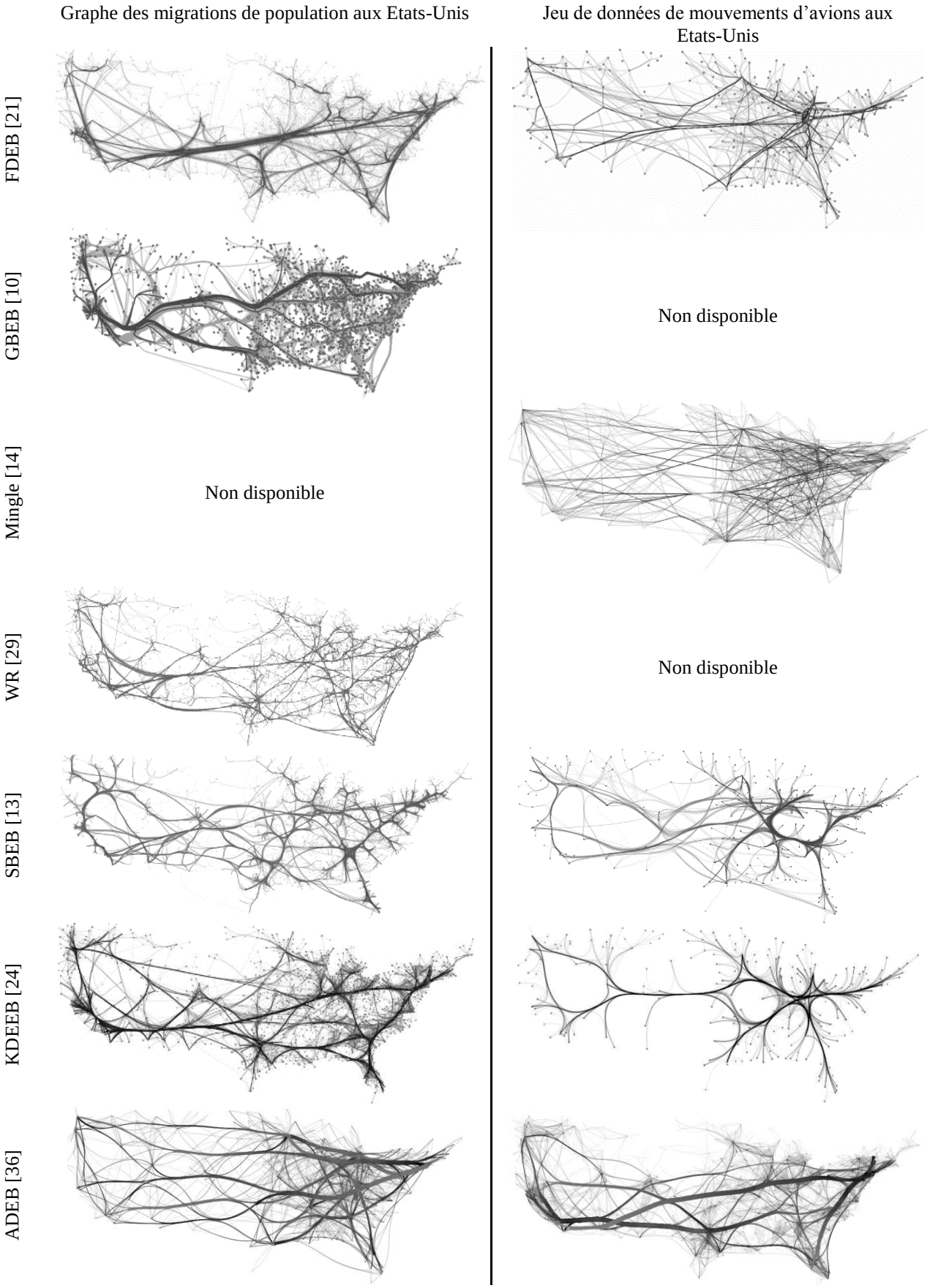
L'advection des liens du graphe s'opère selon deux techniques distinctes.

La première technique, principalement liée aux techniques géométriques, obtient l'advection des nœuds grâce à des B-Spline dont les points de contrôles ont été calculés dans l'étape précédente du pipeline. C'est le cas pour HEB, GDEB et Mingle.

La seconde technique réalise l'advection des liens du graphe par interpolation Eulérienne. Les liens du graphe sont échantillonnés *(resampling)* en lignes brisées. Ensuite, chaque point d'une ligne brisée (à l'exception de ses extrémités) est déplacé par interpolation linéaire à partir d'un vecteur de déplacement défini par la structure de contrôle. Cette technique est utilisée par FDEB, SBEB, KDEEB et ADEB.

Enfin, WR utilise une technique comparable à l'interpolation d'Euler. Chaque lien est échantillonné en ligne brisée à partir d'une implémentation de l'algorithme de Dijkstra. Les liens sont redéfinis comme le plus court chemin entre deux nœuds.

Figure 3. Exemple de visualisation de graphes avec des algorithmes de bundling



Lissage des Liens du Graphe

Réalisé au niveau géométrique, deux types de lissage se distinguent : le lissage par courbes polynomiales (B-Splines ou Bézier) ou par Laplacien d'ordre 1.

Le lissage par B-Spline est privilégié par les techniques suivantes : HEB, GDEB, WR et Mingle. Cette utilisation de B-Spline provient du type de structure de contrôle utilisée. En effet, ces techniques définissent toutes des points de contrôle qui sont alors directement utilisés comme points de contrôle de B-Splines. Les techniques utilisant le Laplacien d'ordre 1 sont ADEB, KDEEB et SBEB.

Par ailleurs, FDEB utilise une technique de lissage personnelle qui regroupe les liens en lignes brisées et applique un produit de convolution entre chaque point d'une ligne brisée (à l'exception des extrémités) et un noyau gaussien.

Techniques d'Amélioration du Rendu Visuel

Sur l'ensemble des techniques de bundling présentées dans la sous-section précédente, toutes utilisent des techniques de traitement d'images afin d'améliorer le rendu visuel. Ces traitements permettent d'améliorer la lisibilité du graphe agrégé ou de coder une information supplémentaire au travers d'un variable visuelle [5]. Ces techniques sont principalement la transparence des liens *(alpha blending)* (figure 5), la gestion de l'occlusion *(z-ordering)* ou encore la gestion de l'éclairage *(shading*.) Dans notre analyse, nous nous sommes concentrés sur les trois étapes d'amélioration du rendu les plus caractéristiques au bundling.

Affichage de la Densité

Le bundling ajoute des espaces vides au détriment de zones plus denses [23]. Ainsi, pour permettre à l'utilisateur de visualiser cette agglomération, les liens agrégés affichent la densité de liens originels présents dans un même regroupement (i.e. la densité du nouveau lien).

Dans l'ensemble des techniques existantes, deux méthodes de représentation de la densité sont utilisées. La première consiste à coder la densité sur une échelle de couleur. La seconde consiste à représenter des liens dont l'épaisseur dépend de la densité. La densité peut aussi être affichée à l'aide de la technique de « bump-mapping » [29][24] (figure 4). HEB utilise une version modifiée du

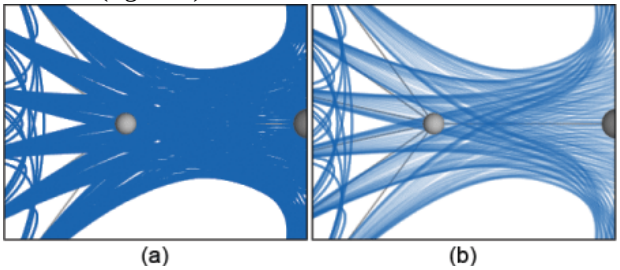


Figure 5. Structure hiérarchique agrégée avec HEB. a) Liens agrégés sans transparence b) Liens agrégés avec gestion de la transparence [22]

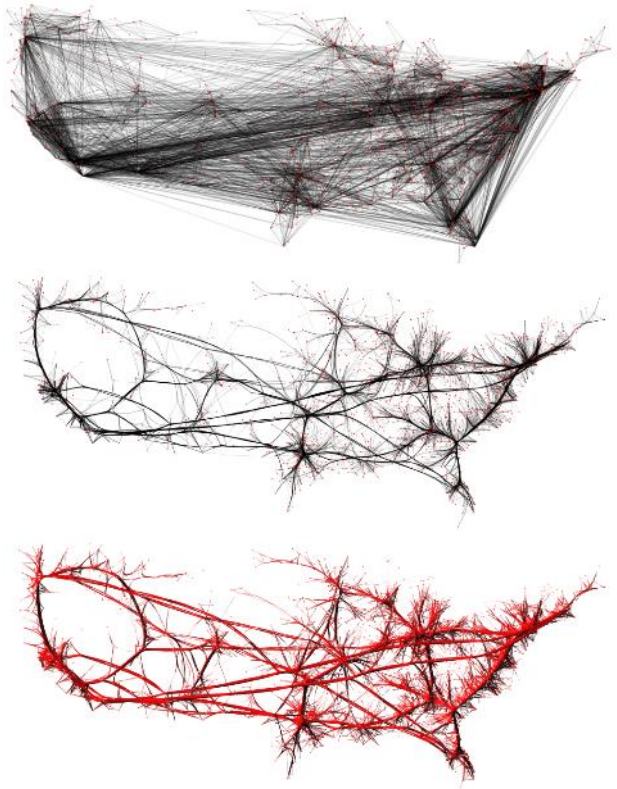


Figure 4. Jeu de données de mouvements d'avions aux Etats-Unis. Version originale en haut, agrégée [24] au milieu, version avec du bump-mapping en bas.

mélange de couleur *(blending)* pour renforcer les différences de densité (figure 5).

Le calcul de la densité est réalisé en post-traitement des données (i.e. après le calcul du graphe agrégé) dans l'ensemble des techniques présentées, à l'exception de KDEEB [24] et ADEB [36]. En effet, ces derniers utilisent la densité dans leurs calculs du graphe agrégé.

Affichage de la Direction

Dans le cas particulier des graphes orientés, plusieurs techniques ont été utilisées pour représenter la direction sous forme de variable visuelle. Ces techniques sont : les dégradés de couleurs, les textures et les animations par particules [36]. Holten et al. ont réalisés des expérimentations permettant de déterminer le choix de design le plus efficace [20][19]. Ces expérimentations montrent que l'animation et la forme (triangle pointant vers la direction) sont les choix les plus pertinents. Récemment, Sheepens et al. [41] ont utilisé un système à particules pour visualiser la direction d'un graphe agrégé.

L'Evitement

Enfin, plus marginal, il est possible d'insérer des obstacles dans un graphe agrégé. Cette méthode permet d'éviter que des liens ne recouvrent des régions particulières [24] du graphe.

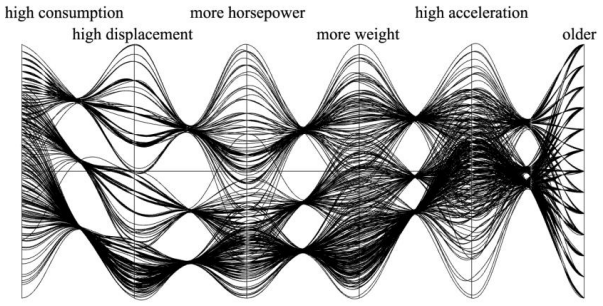


Figure 6. Utilisation du bundling dans la visualisation de coordonnées parallèles [18].

CAS D'UTILISATION

Les algorithmes de bundling sont utilisés dans différents cas d'utilisation.

L'un des premiers cas d'utilisation du bundling apparaît avec HEB [22]. Ce dernier applique le bundling à la visualisation de programme et les graphes hiérarchiques d'appels de fonctions.

SBEB [13], WR [29], KDEEB [24] et ADEB [36] utilisent le bundling pour visualiser des flux. Deux exemples sont très présents dans la littérature ; le graphe des migrations de population aux Etats-Unis ainsi que les trajectoires d'avions sur une journée aux Etats-Unis. Ces deux cas d'utilisation sont visibles en figure 3. De manière similaire, Mingle [14] analyse la répartition des communications internet.

KDEEB [23] et ADEB [26][36] introduisent un nouveau cas d'utilisation en analysant les tracés oculaires mesurés à l'aide d'un capteur oculaire *(eye tracker).* Ce cas d'utilisation exploite l'application du bundling aux graphes orientés (figure 7).

Pupyrev et al. [37] appliquent le bundling pour améliorer la visualisation de graphes de machines à états. Böttger et al. [8] agrègent les liens représentant des connexions neuronales. Enfin, Heinrich et al. [18] implémentent le bundling afin de diminuer le rapport de surface encre/fond dans la visualisation de coordonnées parallèles *(parallel coordinates)* (figure 6).

Bien que la plupart des exemples d'utilisation portent sur l'agrégation de graphes en deux dimensions, certaines techniques permettent l'agrégation dans un espace en trois dimensions [28]. Pour exemple, ce bundling est possible avec HEB (en prenant une B-Spline 3D), FDEB (avec une distance euclidienne en 3D), KDEEB (avec une carte de densité 3D) ou encore SBEB (avec un squelette 3D).

LIMITATIONS, DEFIS ET AXES DE RECHERCHE

La description des différentes implémentations du bundling nous amène à discuter des possibilités qu'elles offrent, de leurs défauts mais aussi des perspectives d'évolutions futures qu'elles suggèrent.

Le Bundling et son Esthétisme

Les algorithmes de bundling ont montré leur capacité à agréger visuellement les liens d'un graphe permettant de libérer des espaces vides dans le but d'améliorer la visibilité. Ce pouvoir d'agrégation permet de faire ressortir plus facilement les informations structurantes d'un jeu de données, comme les flux principaux par exemple. Le caractère esthétique est indéniable avec la production de courbes lisses et l'ajout de traitement d'image comme le bump-mapping. Néanmoins, ce critère d'esthétisme n'a jamais été évalué qualitativement.

Les Limitations du Bundling

Les techniques actuelles de bundling ne permettent pas de répondre à toutes les questions posées par la visualisation de données en général. Et son pouvoir de simplification visuelle a de nombreuses limitations. Dans la suite de cette sous-section, nous classons ses limitations en quatre niveaux :

- les tâches réalisables grâce au bundling,
- les types de jeu de données utilisables,
- le paramétrage des algorithmes,
- la véracité des informations extraites d'un graphe visuellement agrégé.

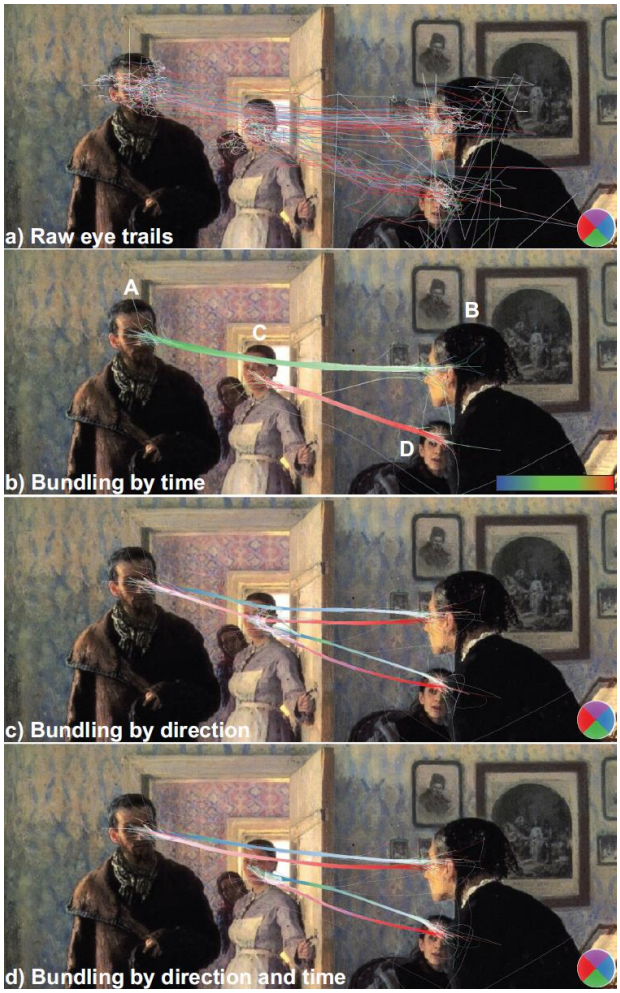


Figure 7. Bundling multicritères d'un jeu de données issu d'un capteur oculaire [36].

Les Tâches

Si l'on fonde notre analyse sur la taxonomie des tâches de graphes introduite par Lee et al. [30], le bundling permet les tâches suivantes :

- suivre un lien ou un chemin, sous certaines conditions de densité réduite et avec des outils d'interaction (pour exemple, glisser le long d'un lien Moscovich et al. [33]),
- la visualisation d'un attribut (ici la densité) d'un lien,
- la reconnaissance de sous-ensembles [31].

Les expérimentations de McGee et al. [31] ont permis d'évaluer deux types de tâches réalisables avec le bundling sur des jeux de données hiérarchiques. Ils montrent que sans ajout de techniques d'interactions, le bundling n'améliore pas la capacité de suivi d'un lien dans un graphe. Néanmoins, il permet de reconnaître plus rapidement des sous-ensembles dans les jeux de données.

Cependant, le bundling ne permet pas simplement de trouver des informations concernant un nombre réduit de liens. Trouver une anomalie (ex. un lien isolé) n'est pas une tâche supportée par le bundling. De même, connaître le nombre de liens d'un nœud n'est pas possible. Enfin, il est impossible de réaliser des tâches impliquant la notion de distance entre deux nœuds avec le bundling comme par exemple trouver le plus court chemin entre deux nœuds.

Parallèlement, en suivant le mantra de B. Schneiderman « Overview first, zoom and filter, then details-on-demand » [43] (i.e. ; Vue d'ensemble en premier, zoom et filtrage puis détails sur demande), le bundling est un outil pour l'exploration macroscopique de graphes. Il permet la visualisation d'ensemble d'un graphe trop dense pour être analysé dans sa globalité. Et l'utilisation de critères d'agrégation (FDEB et ADEB) permet de réaliser une agrégation sélective en fonction de critères de compatibilité.

Les Jeux de Données Analysables

Les techniques de bundling ne s'appliquent pas à tous les jeux de données. Nous distinguons deux catégories de jeux de données selon les techniques présentées dans la section précédente. La première englobe l'ensemble des jeux de données n'évoluant pas temporellement. La seconde prend en compte les jeux de données horodatés qui sont par nature plus complexes en taille et en structure.

Données Statiques

HEB ne permet que le bundling de jeux de données statiques et hiérarchiques. Les autres techniques de bundling étudiées dans cet article (FDEB, SBEB, GBEB, SBEB, KDEEB et ADED) permettent l'analyse de jeux de données statiques non forcément hiérarchisés. Néanmoins, seuls FDEB et ADEB sont capables d'agréger les trajectoires selon des critères définis par l'utilisateur. Les autres techniques présentées agrègent les trajectoires en fonction de la proximité géographique des liens du graphe.

Données Dynamiques

Le passage d'un ensemble de données statiques à celui d'un ensemble de données dynamiques pose des problèmes à la plupart des techniques de bundling. Notamment par le simple besoin d'agréger un graphe rapidement. Les premières techniques à avoir résolu ce problème sont FDEB [35], KDEEB [25] et ADEB [36]. Par ailleurs, les données dynamiques posent aussi des questions de continuité [16] entre les graphes visuellement agrégés d'un même jeu de données. Seuls KDEEB et ADEB, grâce à l'algorithme MeanShift [9], assurent cette continuité.

Le Paramétrage des Algorithmes

Notre expérience acquise sur l'utilisation des algorithmes de bundling nous a montré que le résultat final agrégé dépend des valeurs prises pour les paramètres des algorithmes. Ceci est particulièrement vrai pour les paramètres ayant attraits aux structures de contrôle de l'algorithme, influençant fortement le résultat final. Le graphe final peut ainsi être très fortement ou partiellement agrégé. L'expérience montre aussi qu'il est difficile de déterminer quels sont les bons paramètres sans de nombreux essais préalables.

La Véracité des Informations Visuellement Agrégées

Pour terminer sur les limitations du bundling, le facteur le plus polémique réside dans la véracité de la visualisation graphique [34]. Nous distinguons deux niveaux d'honnêteté de la visualisation : celui de la véracité théorique du graphe agrégé et enfin celui de la véracité aux yeux de l'utilisateur.

Pour ce qui est de la véracité théorique d'un graphe agrégé avec les méthodes de bundling, à ce jour, rien ne peut garantir que le résultat de l'algorithme fasse ressortir une information réellement contenue dans les jeux de données. Par exemple, l'application de KDEEB sur un graphe aléatoire fait ressortir des agrégats de liens. Dans ce cas, deux phénomènes permettent d'expliquer partiellement ce problème. Le premier phénomène est l'impossibilité actuelle de générer un graphe purement aléatoire. Melançon et al. [32] ont réussi à générer aléatoirement un graphe mais dont les liens entre nœuds sont aléatoires. Dans notre cas, c'est la position des nœuds qui est primordiale. Le second, est la difficulté à fournir un

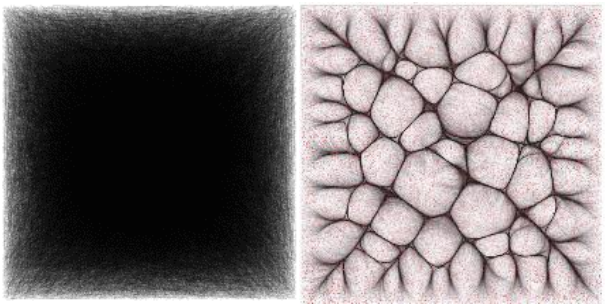


Figure 8. Graphe pseudo aléatoire et sa version agrégée avec KDEEB [24]. Les points rouges représentent les nœuds du graphe.

graphe possédant une densité uniforme de liens (effets de bords d'écran figure 8). Une solution envisageable pour confirmer ou infirmer la véracité d'une technique de bundling pourrait consister à agréger un graphe aléatoire sur une sphère. Cela permettrait ainsi de supprimer les effets de bords.

Dans le cas de la véracité des résultats du bundling aux yeux de l'utilisateur, des pistes de résolution existent. Pour exemple, l'ajout d'animation lors de l'agrégation permettrait aux utilisateurs de visualiser l'agrégation des liens petit à petit [26], renforçant ainsi le sentiment de véracité de la visualisation. Ce type d'implémentation est facilement réalisable avec des algorithmes de bundling itératif (ex : SBEB, FDEB ou encore KDEEB).

Pistes d'Améliorations des Techniques de Bundling

Ces propositions sont issues de notre expérience acquise avec l'utilisation du bundling. Elles dénotent principalement le manque d'outils d'évaluation.

Fournir des Outils d'Evaluation de Liens Agrégés

Aujourd'hui, il n'existe pas de métriques ou d'outils permettant de quantifier le résultat d'un algorithme de bundling. La seule métrique disponible est le rapport de surface encre/fond proposé par Tufte [46], mais qui ne permet pas de qualifier pleinement le résultat d'un algorithme. La figure 3 montre des exemples de résultats d'algorithmes de bundling avec des rendus visuels très différents sur les mêmes jeux de données.

	Pré traitement	**Post traitement**	**Graphe dynamique**	**Graphe orienté**	**Paramètres**	**Complexité**
HEB [22]	Structure hiérarchique	Ajustement de la transparence			B-Spline	O(L)
FDEB [21]		Carte des densités	[35]	[42]	Variable en fonction des critères de compatibilité	$O(L^2)$
GBEB [10]	Maillage de contrôle				B-Spline, maillage de contrôle	O(L.log(L))
WR [29]	Maillage Voronoï Delaunay	Bump mapping			Maillage Voronoï Delaunay, B-Spline	O(L.log(L))
Mingle [14]	Graphe de proximité	Ajustement de la transparence			Graphe de proximité	O(L.log(L))
SBEB [13]	Clustérisation	Renforcement des bords des liens			Squelettisation, lissage	O(L)
KDEEB [24]	Carte de densité	Bump mapping, évitement d'obstacles	[25]		Carte d'accumulation, lissage, attraction	O(L)
ADEB [36]	Carte des densités	Bump mapping, évitement d'obstacles		[36]	Critères de compatibilité, carte d'accumulation, lissage, attraction	O(L)

Table 1. Tableau récapitulatif des techniques de bundling courantes. « L » représente le nombre de liens dans un graphe.

Fournir des Jeux de Données de Références

Chaque nouvelle implémentation d'algorithme de bundling utilise de nouveaux jeux de données. Cela rend difficile les comparaisons entre les différentes techniques. Mingle a utilisé un ensemble de jeux de données de référence [47] mais sans fournir d'indications sur d'éventuelles informations à extraire. Même si dans la littérature, il est possible de voir des comparaisons avec deux jeux de données (trafic aérien aux Etats-Unis et un graphe de migration aux Etats-Unis), nous n'avons pas la connaissance préalable des informations à en extraire pour espérer valider une technique de bundling.

Ainsi, il serait utile de réunir un ensemble de jeu de données validées dont les informations à extraire sont connues. Il serait alors possible de comparer objectivement les résultats des différentes techniques de bundling selon des critères de rapidité de calcul, de qualité du rendu ou encore d'extraction de données de référence.

Relier le Bundling à des Tâches Utilisateur

A travers l'ensemble des techniques abordées dans ce papier, nous pouvons remarquer le manque d'association entre une technique spécifique et une tâche utilisateur. Pour améliorer et démocratiser les techniques de bundling, il est nécessaire de faire correspondre la tâche utilisateur avec l'algorithme le mieux adapté. Le manque de lien entre contraintes techniques d'un algorithme et tâches rend ce travail particulièrement difficile. Pour exemple, il pourrait être intéressant de permettre à l'utilisateur d'agréger deux trajectoires par interaction directe et de laisser l'algorithme spécifier les paramètres nécessaires pour atteindre cet objectif. Dans ce sens, Henry et al. [39] proposent quelques concepts d'agrégation par interaction directe en agençant dynamiquement la position des nœuds d'un graphe.

Améliorer le Paramétrage des Algorithmes

L'ensemble des implémentations actuelles du bundling présente une forte disparité de complexité de paramétrage. HEB est de loin le plus simple, mais aussi le plus contraignant de par la nécessité d'utilisation de structures de données hiérarchiques. Mingle est de loin le plus difficile à paramétrer avec une utilisation du GPU et un calcul complexe du graphe de dépendance dans la structure de contrôle. Tous ces algorithmes utilisent de nombreux paramètres pour leurs calculs. Ces paramètres sont souvent trop abstraits et trop liés à l'algorithme. Pour exemple, KDEEB utilise une carte d'accumulation dont la taille n'a pas de sens en termes de tâches utilisateur.

Fournir des Paramètres de Référence pour les Techniques de Bundling.

Pour s'assurer d'un résultat initialement convenable, il serait important de fournir avec chaque algorithme de bundling un jeu de données de référence adossé avec les paramètres jugés optimaux. Par exemple, KDEEB utilise plusieurs itérations de bundling avec des paramètres qui évoluent à chaque passe. Ceci permettra d'obtenir un résultat satisfaisant, au risque de modifier les paramètres afin de s'adapter à d'autres jeux de données.

Futurs Défis et Opportunités du Bundling

Les techniques de bundling sont utiles à la simplification visuelle de graphes et se sont principalement intéressées aux problèmes liés à la densité d'affichage. Les récentes techniques de bundling utilisent les capacités des cartes graphiques ; soit avec leur capacité d'affichage (OpenGL/WebGL/DirectX), soit avec leur capacité de calcul parallèle (OpenCL/CUDA). Les dernières techniques semblent avoir trouvé une solution aux problèmes de passage à l'échelle avec des graphes de taille importante grâce à l'utilisation des algorithmes basé pixels comme SBEB, KDEEB ou ADEB dans lesquelles la complexité est proche de O(L). Cette simplification des temps de calcul s'explique notamment par la parallélisation massive permise par l'utilisation des processeurs graphiques [27].

A cet instant, ADEB est la seule technique de bundling permettant de prendre en compte plusieurs critères d'agrégations (pour exemple la direction d'un lien ou temporalité d'un lien) avec une complexité proche de O(L).

A ce jour, il n'existe aucuns travaux sur la définition de tâches ou de métriques spécifiques au bundling. En comparaison avec le nombre de techniques de bundling existantes, le travail sur les tâches et les métriques liées au bundling est un des défis de son avenir.

CONCLUSION

Cet article regroupe les techniques de bundling les plus courantes et propose pour la première fois une structure pour les classer et les comparer au moyen d'un pipeline graphique propre aux algorithmes de bundling (figure 2). Cette comparaison s'opère selon des critères techniques qui vont ainsi permettre de choisir au mieux la technique de bundling la plus adaptée à un type de jeu de données. Cet article propose aussi d'identifier les avantages et les défauts des algorithmes de bundling. Enfin, au travers de notre section discussion, nous avons identifié les points limitants de ces techniques et ouvert la voie sur les futures opportunités et défis de l'agrégation visuelle.

REFERENCES

1. Agrawal R. & Srikant R. Fast algorithms for mining association rules. In Proc. *20th int. conf. very large data, VLDB*. Vol 1215, (1994), 487–499.
2. Amar R., Eagan J. & Stasko J. Low-level components of analytic activity in information visualization. In *Information Visualization, 2005*, IEEE (2005), 111–117.
3. Barsky B.A. The Beta-spline: a local representation based on shape parameters and fundamental geometric measures. (1981).
4. Beck F., Burch M., Diehl S. & Weiskopf D. The state of the art in visualizing dynamic graphs. *EuroVis STAR*, (2014).
5. Bertin J. *Graphics and graphic information processing*. Walter de Gruyter, 1981.
6. Blinn J.F. Simulation of wrinkled surfaces. In *ACM SIGGRAPH computer graphics* (1978), 286–292.
7. Bothorel G., Serrurier M. & Hurter C. Visualization of frequent itemsets with nested circular layout and bundling algorithm. In *Advances in Visual Computing*. Springer (2013), 396–405.

8. Bottger J., Schafer A., Lohmann G., Villringer A. & Margulies D.S. Three-dimensional mean-shift edge bundling for the visualization of functional connectivity in the brain. *Visualization and Computer Graphics, IEEE Transactions on 20*, 3 (2014), 471–480.
9. Comaniciu D. & Meer P. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 5 (2002), 603–619.
10. Cui W., Zhou H., Qu H., Wong P.C. & Li X. Geometry-based edge clustering for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on 14*, 6 (2008), 1277–1284.
11. Dickerson M., Eppstein D., Goodrich M.T. & Meng J.Y. Confluent drawings: visualizing non-planar diagrams in a planar way. In *Graph Drawing*, Springer (2004), 1–12.
12. Dwyer T., Marriott K., & Wybrow M. Integrating edge routing into force-directed layout. In *Graph Drawing*, Springer (2007), 8–19.
13. Ersoy O., Hurter C., Paulovich F.V., Cantareiro G., & Telea A. Skeleton-based edge bundling for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2364–2373.
14. Gansner E.R., Hu Y., North S., & Scheidegger C. Multilevel agglomerative edge bundling for visualizing large graphs. In *PacificVis 2011*, IEEE (2011), 187–194.
15. Gansner E.R. & Koren Y. Improved circular layouts. In *Graph Drawing*, Springer (2007), 386–398.
16. Ghani S., Elmqvist N., & Yi J.S. Perception of Animated Node-Link Diagrams for Dynamic Graphs. In *Computer Graphics Forum*, Blackwell Publishing Ltd (2012), 1205–1214.
17. Haber R. B. & McNabb D. A. Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing*, 1990, 74, 93
18. Heinrich J., Luo Y., Kirkpatrick A.E., Zhang H., & Weiskopf D. Evaluation of a bundling technique for parallel coordinates. *arXiv preprint arXiv:1109.6073*, (2011).
19. Holten D., Isenberg P., Van Wijk J.J., & Fekete J.-D. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *PacificVis 2011*, IEEE (2011), 195–202.
20. Holten D. & van Wijk J.J. A user study on visualizing directed edges in graphs. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2009), 2299–2308.
21. Holten D. & Van Wijk J.J. Force-Directed Edge Bundling for Graph Visualization. In *Computer Graphics Forum*, Blackwell Publishing Ltd (2009), 983–990.
22. Holten D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (2006), 741–748.
23. Hurter C., Ersoy O., Fabrikant S.I., Klein T.R. & Telea A.C. Bundled Visualization of DynamicGraph and Trail Data. *Visualization and Computer Graphics, IEEE Transactions on 20*, 8 (2014), 1141–1157.
24. Hurter C., Ersoy O. & Telea A. Graph bundling by kernel density estimation. In *Computer Graphics Forum*, Blackwell Publishing Ltd (2012), 865–874.
25. Hurter C., Ersoy O. & Telea A. Smooth bundling of large streaming and sequence graphs. In *PacificVis 2013*, IEEE (2013), 41–48.
26. Hurter C., Telea A. & Ersoy O. Moleview: An attribute and structure-based semantic lens for large element-based plots. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2600–2609.
27. Hurter C. Toward image based algorithm to support interactive data exploration. Thèse de doctorat (2014).
28. Lambert A., Bourqui R. & Auber, D. 3D edge bundling for geographical data visualization. In *Information Visualisation (IV), 2010 14th International Conference* IEEE (2010), 329–335.
29. Lambert A., Bourqui R. & Aube D. Winding roads: Routing edges into bundles. In *Computer Graphics Forum*, Blackwell Publishing Ltd (2010), 853–862.
30. Lee B., Plaisant C., Parr C.S., Fekete J.-D. & Henry, N. Task taxonomy for graph visualization. In *Proc. of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, ACM (2006), 1–5.
31. McGee F. & Dingliana J. An empirical study on the impact of edge bundling on user comprehension of graphs. In *Proc. of the International Working Conference on Advanced Visual Interfaces*, ACM (2012), 620–627.
32. Melançon G., Dutour I. & Bousquet-Mélou M. Random generation of directed acyclic graphs. *Electronic Notes in Discrete Mathematics 10*, (2001), 202–207.
33. Moscovich T., Chevalier F., Henr, N., Pietriga E. & Fekete J.-D. Topology-aware navigation in large networks. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2009), 2319–2328.
34. Nguyen Q., Eades P. & Hong S.-H. On the faithfulness of graph visualizations. In *PacificVis 2013*, IEEE (2013), 209–216.
35. Nguyen, Q., Eades, P., and Hong, S.-H. StreamEB: Stream Edge Bundling. In *Graph Drawing*, Springer (2013), 400–413.
36. Peysakhovich V., Hurter C. & Telea A. Attribute-Driven Edge Bundling for General Graphs with Applications in Trail Analysis. In *proc. PacificVis 2015*.
37. Pupyrev S., Nachmanson L. & Kaufmann, M. Improving layered graph layouts with edge bundling. In *Graph Drawing;* Springer (2011), 329–340.
38. Qu H., Zhou H. & Wu Y. Controllable and progressive edge clustering for large networks. In *Graph Drawing*, Springer (2007), 399–404.
39. Riche N.H., Dwyer T., Lee B. & Carpendale S. Exploring the design space of interactive link curvature in network diagrams. *In Proc. of the International Working Conference on Advanced Visual Interfaces*, ACM (2012), 506–513.
40. Schaeffer S.E. Graph clustering. *Computer Science Review 1*, 1 (2007), 27–64.
41. Scheepens R., Hurter C., Van der Wetering H. & Van Wijk J.J. Visualization, Selection, and Analysis of Traffic Flows. In *Proc InfoVis 2015*
42. Selassie D., Heller B., and Heer J. Divided edge bundling for directional network data. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2354–2363.
43. Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages 1996*, IEEE (1996), 336–343.
44. Shulz H, Hurter C, Grooming the Hairball – How to tidy up network visualizations? *Tutorial Infovis 2013*.
45. Tamassia R. *Handbook of graph drawing and visualization*. CRC press, 2013.
46. Tufte E.R. & Graves-Morris P. *The visual display of quantitative information*. Graphics press Cheshire, CT, 1983
47. University of Florida Sparse Matrix Collection. http://www.cise.ufl.edu/research/sparse/matrices/
48. Voronoi G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik 133*, (1908), 97–178.
49. Zhou H., Yuan X., Cui W., Qu H. & Chen, B. Energy-based hierarchical edge clustering of graphs. In *PacificVis 2008*, IEEE (2008), 55–61.