REBANKMENT : UN ALGORITHME POUR DÉPLACER LES TALUS SUR LES CARTES PAR MOINDRES CARRÉS

par Guillaume Touya et Imran Lokhat

LASTIG, Univ Gustave Eiffel, IGN-ENSG
F-77420 Champs-sur-Marne, France
Guillaume.touya@ign.fr, Imran.lokhat@ign.fr

Même si les progrès récents en automatisation de la généralisation cartographique aident les agences de cartographie nationales à produire leurs cartes topographiques à différentes échelles de plus en plus rapidement, il existe encore des opérations de généralisation que nous ne savons pas automatiser correctement. Par exemple, les talus sont fréquemment représentés par un symbole linéaire avec des barbules représentant le sens de la pente du talus. Ce type de symbole prend de la place et nécessite d'être éloigné des symboles de routes notamment. Cet article propose un algorithme, appelé ReBankment, qui permet de déplacer automatiquement les lignes de talus. L'algorithme utilise une triangulation pour identifier les voisinages entre objets de la carte, puis une optimisation par moindres carrés de la position des points de la ligne de talus, ce qui permet un déplacement sans modifier la forme initiale de la ligne. L'article propose également des moyens pour traiter les cas complexes et les jeux de données massifs. L'algorithme est testé sur des données réelles de l'IGN France pour la généralisation de la carte au 1 : 25000.

Mots-clés : *généralisation cartographique, moindres carrés, déplacement, carte topographique, talus*

While the recent progress on automated generalisation helped National Mapping Agencies to derivetopographic maps more and more quickly, there are still practical cartographic issues that requireattention. For instance, embankments are represented with line symbols showing the slope of theembankment. This paper proposes an automated algorithm, called ReBankment that displaces theembankment lines from the roads and rivers that overlap the embankment symbol. ReBankment isbased on a triangulation to identify neighbourhoods, and on a least squares adjustment to displaceand distort the embankment line while preserving its shape. The paper also proposes how to handle complex cases and scaling issues. ReBankment is tested on real data from a 1:25k scale topographicmap.

Keywords **:** *map generalisation, least squares, displacement, topographic map, embankment*

Introduction

Les cartes topographiques, en particulier celles au 1/25000, se doivent de proposer une représentation précise et compréhensible du relief, car ces cartes sont utilisées pour la randonnée, les services de secours, ou les levés de précision. Par conséquent, l'IGN utilise des courbes de niveau, un relief ombré, et des points cotés pour bien montrer les formes du relief. Les talus sont des formes spécifiques du terrain qui sont souvent trop petites et localisées pour être visibles avec les courbes de niveau. Mais ces talus sont importants car ils sont souvent des obstacles sur le terrain, donc on choisit souvent de les représenter avec des symboles linéaires spécifiques, auxquels on ajoute de petites barbules qui donnent le sens de la pente du talus. Comme on peut le voir sur la partie droite de la figure 1, un tel symbole a un encombrement important, notamment parce que l'on trouve souvent les talus le long des routes ou des cours d'eau, qui sont eux représentés avec des symboles assez larges, comme on peut le voir sur la partie gauche de la figure 1 où les lignes de talus sont cachées derrière les symboles des routes et des rivières. C'est pourquoi il est nécessaire de déplacer les symboles des talus, et parfois de les compresser, pour leur rendre leur lisibilité dans la carte.

Les agences de cartographie nationales, ou régionales, comme l'IGN, ont maintenant des rythmes de mise à jour de leur carte très rapides, ce qui pousse la production de ces cartes à être la plus automatisée possible. Si la carte au 1 : 25000 doit être publiée en ligne deux à trois fois par an, on ne peut pas se permettre de généraliser les talus à chaque fois à la main, le processus se doit d'être totalement automatique. Notre objectif est donc ici de viser un algorithme qui fournit des résultats satisfaisants sans reprise interactive pour corriger les défauts.

La plupart des algorithmes de déplacements proposés dans la littérature en généralisation cartographique sont dédiés au déplacement d'éléments rigides, comme des bâtiments, qui doivent être déplacés d'un bloc, sans déformation, par une simple translation. Par conséquent, il est nécessaire de développer un nouvel algorithme, capable de réaliser des déplacements élastiques, qui déforment légèrement, ou compressent les talus afin d'éviter les collisions avec les éléments de la carte entourant le talus, et afin de préserver au mieux la forme du talus. Faire en sorte que cet algorithme puisse être rapidement disponible pour les chaînes de production cartographique de l'IGN est un autre défi, qui a des conséquences à la fois sur la conception de l'algorithme et son implémentation. Dans cet article nous présentons donc l'algorithme ReBankment pour répondre à ces deux défis. Il se fonde sur le principe de la généralisation par moindres carrés (Harrie, 1999 ; Sester, 2000), et nous avons proposé des contraintes spécifiques à la généralisation des talus, ainsi que des pré-traitements permettant de gérer de nombreuses situations complexes.

La suite de l'article reprend le plan suivant. La deuxième partie présente l'état de l'art sur le déplacement automatique en généralisation cartographique et sur l'utilisation des moindres carrés. Ensuite, la partie 3 décrit en détail l'algorithme ReBankment avec notamment la présentation de tous les pré-traitements. Les résultats et leur évaluation sont présentés en partie 4, avant une conclusion et une discussion des perspectives de ce travail.

Etat de l'art

Quand l'échelle diminue, l'emprise des symboles de la carte augmente proportionnellement par rapport à la taille réelle des objets sur le terrain, ce qui entraîne mécaniquement des superpositions de symboles pour les objets proches les uns des autres, et le déplacement des objets est considéré depuis très longtemps comme une opération indispensable de la généralisation cartographique (Brassel et Weibel, 1988). Pour déplacer des éléments linéaires de réseau comme des routes, des rivières ou des voies ferrées, la première idée a été de trouver les zones de conflits (là où les superpositions de symboles ont lieu), afin de déplacer localement les lignes (Nickerson, 1988). Mais ce processus est complexe, et le déplacement automatique a beaucoup progressé avec l'introduction de méthodes d'optimisation de ces déplacements locaux, que ce soit avec des méthodes comme les *Snakes* (Burghardt et Meier, 1997), ou plus tard les *poutres élastiques* (Bader et Barrault, 2000, Bader *et al.*, 2005).

Mais les recherches sur le déplacement en généralisation cartographique ont essentiellement porté sur le déplacement de bâtiments ou de symboles ponctuels en zone urbaine (Mackaness, 1994), car les bâtiments ont souvent été le thème majeur des recherches en généralisation. Comme pour les routes, les premières méthodes ont adopté une approche séquentielle, déplaçant itérativement chaque bâtiment jusqu'à ce que chaque conflit soit résolu (Ruas, 1998). Plusieurs algorithmes plus récents reprennent cette approche itérative en la perfectionnant, ou en la spécialisant (Kovanen *et al.*, 2013 ; Wei *et al.*, 2018 ; Liu *et al.*, 2019). D'autres approches, plus globales, ont été proposées, comme le calcul d'un champ vectoriel de déplacement pour tout un îlot urbain afin de préserver au mieux les relations spatiales d'alignement entre bâtiments (Ai *et al.*, 2015). Et encore une fois, comme pour les routes, des algorithmes utilisant des méthodes d'optimisation ont été proposés, par exemple en suivant les mêmes principes que les Snakes (Hojholt, 2000), ou que les poutres élastiques (Liu *et al.*, 2014), ou utilisant d'autres méta-heuristiques d'optimisation comme les algorithmes génétiques (Sun *et al.*, 2016).

En ce qui concerne la généralisation de lignes représentant des formes naturelles du relief, comme les talus, ou les lignes de crête, peu d'algorithmes ont été proposés. Le modèle de généralisation GAEL (Gaffuri, 2007) peut être utilisé pour réaliser des déplacements élastiques de telles lignes (Gaffuri, 2009). Et dans le passé, l'IGN a utilisé une adaptation des poutres élastiques pour déplacer les talus dans une carte topographique au 1 : 25000, mais les résultats nécessitaient un certain nombre de reprises interactives pour en corriger les défauts (Lecordix *et al.*, 2007).

Dans le même temps, la généralisation par moindres carrés a démontré sa flexibilité pour des tâches de généralisation dans lesquelles il faut optimiser de petites déformations, comme c'est le cas dans notre problème de déplacements des talus. Les principes de la généralisation par moindres carrés ont été proposés simultanément par Lars Harrie (Harrie, 1999) et Monika Sester (Sester, 2000) il y a un peu plus de 20 ans. Ils ont depuis été étendus pour s'adapter à la généralisation de divers types d'objets d'une carte (Harrie et Sarjakoski, 2002 ; Harrie, 2003 ; Sester, 2005). Les principes de déformation par moindres carrés ont même été adaptés à des problèmes connexes à la généralisation cartographique comme le recalage géométrique de données hétérogènes (Gruendig *et al.*, 2007 ; Dalyot *et al.*, 2012, Touya *et al.*, 2013), le morphing de polylignes (Peng *et al.*, 2013), les loupes interactives (van Dijk et Haunert, 2014), ou encore

l'équarrissage de polygones (Lokhat et Touya, 2016). Ces applications réussies et la maturité de l'approche par moindres carrés nous ont convaincu de l'utiliser pour résoudre notre défi de déplacement automatique élastique des talus dans une carte topographique.

L'algorithme ReBankment

L'algorithme ReBankment, qui déforme et déplace les lignes représentant des talus, est présenté en détail dans cette partie. Nous détaillons d'abord les principes de la généralisation par moindres carrés, puis nous présentons les contraintes proposées spécifiquement pour les talus.

Principes du déplacement par moindres carrés

Comme nous l'avons expliqué dans la partie précédente, les principes de la généralisation par moindres carrés ont été proposés simultanément par Lars Harrie (Harrie, 1999 ; Harrie et Sarjakoski, 2002) et Monika Sester (Sester, 2000, 2005). L'idée est de traduire les contraintes contradictoires de la généralisation cartographique (simplifier les formes tout en préservant la forme initiale) en un système d'équations linéaires, surcontraint et contenant donc plus d'équations que d'inconnues. Ce système n'a donc pas de solution exacte, et la méthode des moindres carrés permet d'obtenir une solution approchée optimale. L'équation (1) montre comment une contrainte de généralisation peut être traduite en une fonction linéaire f dont les inconnues sont la différence entre les coordonnées (xi, yi) du point i d'une géométrie à généraliser, avant et après la généralisation.

$$f(x_1, y1, \ldots, x_n, y_n) = c_{1,1} \cdot \Delta x_1 + c_{1,2} \cdot \Delta y_1 + \cdots + c_{n,1} \cdot \Delta x_n + c_{n,2} \cdot \Delta y_n + c \tag{1}$$

Où $\Delta x_i = x_i - x_i^0$, x_i étant l'abscisse du ième point après généralisation et x_i^0 l'abscisse du ième point avant la généralisation.

Certaines contraintes de généralisation, comme les contraintes sur une distance, ou sur un angle, ne peuvent pas être directement traduites par une fonction *f* linéaire, et dans ce cas, nous réalisons une approximation linéaire de cette fonction *f* par le polynôme de Taylor de premier ordre (équation 2).

$$\frac{\partial f}{\partial x_1} \cdot \Delta x_1 + \frac{\partial f}{\partial y_1} \cdot \Delta y_1 + \frac{\partial f}{\partial x_2} \cdot \Delta x_2 + \frac{\partial f}{\partial y_2} \cdot \Delta y_2 + \cdots + \frac{\partial f}{\partial x_n} \cdot \Delta x_n + \frac{\partial f}{\partial y_n} \cdot \Delta y_n = c - f(\vec{x^0}) \tag{2}$$

Où le vecteur X^0 est composé des coordonnées initiales des points composant la géométrie contrainte par *f*. Quand on assemble toutes ces équations avec toutes les contraintes, sur toutes les géométries d'une portion de carte, on obtient le système d'équation présenté en notation matricielle dans l'équation (3).

$$\vec{A} \cdot \vec{x} = \vec{l} + \vec{v} \tag{3}$$

Où A est la matrice Jacobienne du système, contenant les valeurs ci,j de l'équation (1), X est le vecteur de toutes les inconnues Δxi, L est le vecteur contenant les valeurs constantes notées c dans l'équation (1), et V est le vecteur des résidus, qui contient les erreurs de la solution trouvée par rapport à chaque contrainte. La solution optimale de l'équation (3) est celle donnée dans l'équation (4).

$$\vec{x} = (\vec{A^T} \cdot \vec{P} \cdot \vec{A})^{-1} \cdot \vec{A^T} \cdot \vec{P} \cdot \vec{l} \tag{4}$$

Où P est une matrice diagonale qui contient les poids relatifs de chacune des contraintes.

Comme beaucoup des contraintes ne sont pas linéaires et que le polynôme de Taylor n'est pas une approximation parfaite, la résolution du système contenant ces équations linéarisées n'est pas toujours optimale. C'est pourquoi nous choisissons une approche itérative, et nous résolvons l'équation (4) autant de fois que nécessaire pour que les talus soient assez déplacés. Après chaque itération, le vecteur des positions initiales des géométries (X dans l'équation (3)) devient en fait le vecteur des positions après l'itération précédente, ce qui fait que les talus sont en fait déplacés petit à petit à chaque itération. L'algorithme converge quand il atteint un nombre d'itérations maximum que nous avons fixé, ou dans la plupart des cas, quand le déplacement après une itération devient négligeable.

Contraintes sur les talus

Nous avons défini 4 contraintes pour permettre un déplacement optimal des talus par rapport aux routes (les routes sont fixes et ne sont pas déplacées par l'algorithme) tout en préservant la forme initiale de ces talus :

- (C_1) contrainte de préservation de la position
- (C_2) contrainte de préservation de la forme
- (C_3) contrainte de non superposition des symboles
- (C_4) contrainte de position relative entre talus (Duchêne *et al.*, 2012)

Ces quatre contraintes et leur traduction sous forme d'équation sont présentées dans les sous-parties suivantes.

Contrainte de préservation de la position

La contrainte (C_1) est tout fait usuelle en généralisation par moindres carrés : les objets généralisés ne doivent pas bouger trop loin de lors position initiale dans la carte, qui représente leur position réelle sur le terrain. Nous utilisons la même manière de traduire cette contrainte sous la forme d'équations linéaires que celle proposée par Harrie (1999) et Sester (2000) : les coordonnées des points devraient rester les mêmes (équation 5). Comme les contraintes ne sont pas complètement satisfaites, les points vont bien pouvoir être déplacés un peu dans la solution optimale, mais cette contrainte garantit que cette solution optimale ne soit pas trop éloignée de la position initiale.

$$\begin{cases} \Delta x_i = 0 \\ \Delta y_i = 0 \end{cases} \tag{5}$$

Contrainte de préservation de la forme

La contrainte (C_2) a pour objectif de préserver les formes courbes ou droites des talus avant la généralisation. Cette contrainte est dérivée de plusieurs contraintes similaires proposées par Harrie (1999). Comme le montre la figure 2, cette contrainte est appliquée à chacun des couples de segments consécutifs d'une ligne et contraint à la fois l'angle incident entre ces deux segments (α dans la figure 2) à rester le même, ce qui se traduit par l'équation (6), et contraint les deux segments à garder leur longueur durant la généralisation, ce qui se traduit par l'équation (7).

$$\begin{cases} \| \vec{u} \wedge \vec{w} \| = \| \vec{u} \| \cdot \| \vec{v} \| \cdot \sin \alpha \\ \| \vec{U} \wedge \vec{W} \| = \| \vec{U} \| \cdot \| \vec{W} \| \cdot \sin \beta \end{cases} \tag{6}$$

Toutes ces équations sont non-linéaires, ce qui signifie que l'on doit les approximer par le polynôme de Taylor du premier ordre, en dérivant ces différentes équations, avant de les intégrer au système d'équations global.

$$\begin{aligned} \| w \| = \| W \| \Leftrightarrow & \sqrt{(x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2} \\ & - \sqrt{(x^0_{i+1} - x^0_i)^2 - (y^0_{i+1} - y^0_i)^2} = 0 \end{aligned} \tag{7}$$

Contrainte de non superposition des symboles

Avec la contrainte (C_3), nous voulons que les lignes de talus restent à une distance des routes suffisant à l'affichage du symbole des routes, de celui des talus, avec en plus un petit espace blanc pour permettre à l'œil de bien distinguer les deux symboles. L'écart

$distmin_k$ entre les deux lignes doit donc être de $largeur_i$ + $largeur_k$ + seuil_séparation, où $largeur_i$ est la moitié de la largeur du symbole du talus i, $largeur_k$ est la moitié de la largeur du symbole de la route k, et seuil_séparation est la distance nécessaire pour que l'œil distingue la séparation (0,1 mm sur la carte d'après les études sur le sujet). Comme les symboles de routes varient en fonction de leur importance, cette valeur de $distmin_k$ varie elle aussi entre les routes. L'équation (8) montre comment la contrainte de non superposition des symboles est calculée en fonction de cette valeur $distmin_k$.

$$\mathrm{dist}(v_i, r_k) = \max\left(\mathrm{distmin}_k - \mathrm{dist}(v_i^0, r_k), 0\right) \tag{8}$$

Où v_i est le ième point d'un talus, et v_i^0 est sa position initiale. Cette équation doit être calculée théoriquement pour chaque couple de points d'un talus et de route, ce qui donne un trop grand nombre de couples possibles. Pour limiter ce nombre, nous ne contraignons la non superposition d'un talus qu'avec les routes qui sont juste autour de lui (voir la partie « Passage à l'échelle » pour plus de précisions). Puisque le calcul de la distance entre un point et un segment ne peut pas se faire de manière linéaire, cette équation est une fois de plus approximée par le polynôme de Taylor de premier ordre.

Contrainte de position relative entre talus

La dernière contrainte, (C_4), a pour objectif de préserver la position relative entre deux talus proches, quand un des deux est déplacé du fait de sa proximité avec une route. Comme d'autres contraintes proposées dans les recherches passées sur la généralisation par moindres carrés, cette contrainte utilise une triangulation de Delaunay contrainte entre tous les points des talus pour déterminer quels sont les points qui devraient être contraints. Ce sont certains des segments de cette triangulation qui sont contraints avec une contrainte qui force ces segments à garder la même longueur, comme dans l'équation (7). La figure 3 montre comment nous classons les segments de la triangulation contrainte, pour choisir lesquels doivent être contraints. Nous définissons cinq types de segments :

- *Segment de type 1* : ce sont les segments qui connectent deux points du même talus. Il n'y a aucune raison de contraindre ces segments donc ils sont ignorés.
- *Segment de type 2* : ce sont les segments qui constituent un morceau d'un talus. Ces segments sont déjà contraints par (C_2), donc il n'est pas nécessaire de le faire une deuxième fois et ils sont ignorés.
- *Segment de type 3* : ce sont les segments qui connectent des points appartenant à deux talus différents, avec une longueur supérieure au seuil de distance minimum entre les deux talus, mais inférieur à un seuil d'influence. Ces segments sont contraints pour garder la même longueur.
- *Segment de type 4* : ce sont les segments qui connectent les points de deux talus différents, mais avec une longueur inférieure au seuil de distance minimale entre talus. Ces segments sont contraints pour avoir une longueur finale égale à ce seuil de distance minimale.
- *Segment de type 5* : ce sont les segments qui connectent les points de deux talus différents, mais cette fois avec une longueur supérieure au seuil d'influence entre talus. Ces segments ne sont pas contraints.

Comme (C_3), la contrainte (C_4) peut générer des calculs trop complexes si la triangulation contrainte est calculée sur un trop grand nombre de talus en même temps. C'est pourquoi nous ne calculons la triangulation de Delaunay que sur de petites zones dédiées (voir la partie sur le passage à l'échelle pour plus de détails).

Pondération des contraintes

Comme le montre l'équation (4), une matrice de poids est utilisée pour obtenir la solution optimale, pour pouvoir introduire des importances différentes entre chaque contrainte. Pour la généralisation par moindres carrés, trouver le jeu de poids idéal est souvent un exercice complexe (Harrie, 2003), et c'est encore le cas ici. La table 1 récapitule les poids que nous avons déterminés de manière empirique, pour obtenir nos meilleurs résultats. Nous avons fait une analyse de sensibilité en faisant varier chacun des poids individuellement pour obtenir ces valeurs. (C_3) a de loin le poids le plus important, et cela traduit l'importance de cette contrainte qui est celle qui initie la généralisation : nous voulons éloigner les talus des routes en priorité.

Contrainte	C_1	C_2	C_3	C_4
Poids	8	8	200	15

Tableau 1 : Poids optimaux pour chaque contrainte, déterminés de manière empirique.

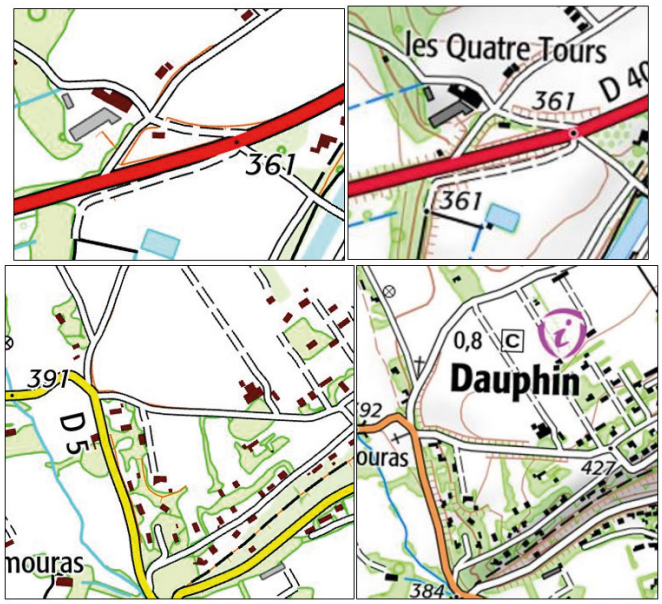

Figure 1 : Deux extraits de cartes topographiques IGN, sans déplacement de talus à gauche et avec un processus de déplacement semi-automatique à droite

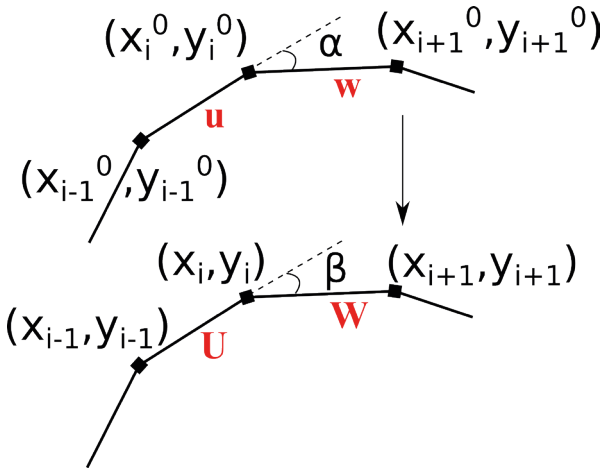

Figure 2 : Une polyligne à déplacer en haut, et sa version déplacée en dessous. La contrainte de préservation de forme force les angles α et β à rester identiques, et les normes des vecteurs u et U (resp. w et W) à rester identiques.

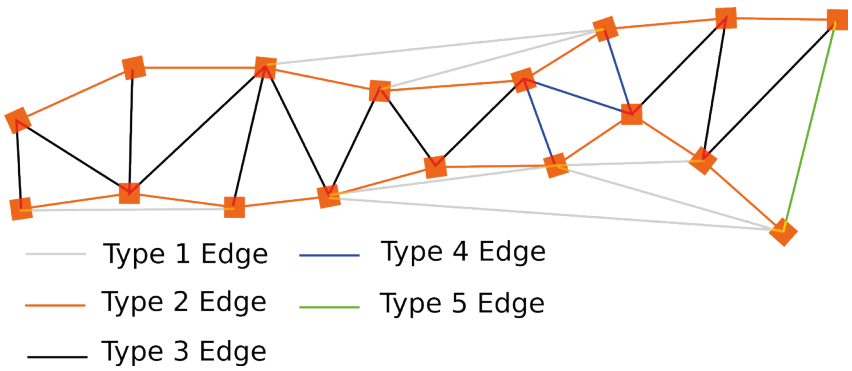

Figure 3 : La triangulation contrainte utilisée pour contraindre la position relative de plusieurs talus s'appuie sur une classification des segments de la triangulation : les segments de type 1 connectent deux points du même talus ; les segments de type 2 font partie du tracé du talus ; les segments de type 3 connectent deux talus différents avec une longueur supérieure au seuil de distance minimale mais inférieure à un seuil d'influence (plus grand que le seuil de distance minimale) ; les segments de type 4 connectent deux talus avec une longueur inférieure au seuil de distance minimale ; enfin, les segments de type 5 connectent deux talus avec une longueur supérieure au seuil d'influence.

Passage à l'échelle

Les techniques de généralisation par moindres carrés nécessitent l'inversion d'une grande matrice (voir équation (4), et c'est une opération qui est très gourmande en ressources informatiques, donc le nombre d'objets que l'on peut traiter en une fois est relativement limité (Touya *et al.*, 2013). De manière générale, la généralisation cartographique est de toute façon une opération nécessitant des calculs très longs, et il est très fréquent de devoir partitionner les données pour éviter de devoir tout généraliser d'un coup (Touya *et al.*, 2017). Dans ce cas, les routes et les rivières forment des frontières naturelles pour nos talus, car les talus ne sont pas supposés traverser ces éléments. Nous proposons de construire un maillage des données à partir des routes et des rivières, intégrées dans un graphe planaire.

La figure 4 montre trois des plus grandes mailles de notre zone d'étude (voir la partie sur les résultats pour la description de ces données). La figure montre que cela permet clairement de limiter le nombre de talus traités dans une maille. Par contre, même si le nombre de talus est modéré, le nombre de routes et de rivières potentiellement en conflit avec ces talus peut rester trop grand pour être traité dans une unqiue matrice, dans ces grandes mailles particulièrement. Donc pour générer des zones de travail encore plus petites, nous consruisons des regroupements de talus proches dans une même maille, ces groupes étant séparés par de grands espaces vides dans la maille (figure 4). Ces regroupements sont fait en reprenant des idées déjà éprouvées en généralisation (Touya, 2010) : (1) on construit une zone tampon autour de chaque talus avec une valeur de trois fois la distance minimale entre deux talus ; (2) on fusionne toutes les zones tampons qui s'intersectent ; (3) on ajoute à notre zone toutes les routes et rivières qui intersectent cette fusion des zones tampons.

Cas complexes

Le fait de travailler sur des jeux de données réels, à l'échelle d'un pays grand comme la France, cause des difficultés nouvelles. Dans notre cas, tester l'algorithme sur des données réelles nous a montré un certain nombre de problèmes dans les données qui empêchent la bonne application de l'algorithme :

- des lignes de talus connectées entre elles ;
- des talus qui touchent des routes (ou des cours d'eau) ;
- des situations sans bonne solution de déplacement par manque d'espace libre ;
- d'autres erreurs dans les données initiales (talus qui se croisent, ou croisent une route).

Dans les données de notre cas d'étude, nous avons trouvé qu'environ 5% des talus représentaient un des quatre cas complexes précédents. La figure 5 montre plusieurs exemples du cas complexe (1), c'est-à-dire des lignes de talus connectées entre elles, ou plutôt des talus représentés par plusieurs lignes au lieu d'une seule. Dans ces situations, utiliser notre algorithme sur les données brutes génère souvent des déconnexions, et il est donc préférable d'ajouter un pré-traitement qui fusionne ces lignes connectées avant de les généraliser.

Le cas complexe (2) est typiquement dû au processus d'abstraction réalisé par les opérateurs qui ont saisi ces lignes par stéréo-restitution : des talus proches des routes sont étendus jusqu'à la ligne représentant la route, et les talus parallèles et très proches de la route sont saisis en partage de géométrie avec la route (figure 6). C'est un problème car notre contrainte de non superposition entre talus et routes ne peut pas gérer les partages de géométrie. Nous commençons par identifier ces deux cas complexes en qualifiant les relations topologiques entre tous les talus et toutes les routes proches. Quand un talus a une relation topologique de type « toucher » ou « superposer » (Egenhofer *et al.*, 1994) avec une route (ou un cours d'eau), il est alors pré-traité en étant déplacé très légèrement de la route, juste assez pour qu'il soit disjoint de la route (figure 7).

Dans le cas d'une relation « toucher », un vecteur de translation est calculé en utilisant l'orientation du premier segment du talus, avec une norme de 1 m. Dans le cas d'une relation « superposer », l'orientation de la partie superposée est calculée et le talus subit une translation d'orientation orthogonale et de norme 1 m'encore une fois. Le cas complexe (3) est classique en généralisation, car les symboles de routes ou de cours d'eau sont souvent larges dans les cartes topographiques, et ils prennent donc beaucoup de place, laissant même parfois un espace vide trop petit pour afficher les éléments voisins (figure 8). Afin de déterminer l'espace libre autour de chaque talus, nous proposons d'utiliser une approche similaire à celle de CartACom (Duchêne *et al.*, 2012). Une zone tampon est calculée autour du talus, et les polygones créés par les symboles des routes sont enlevés dans cette zone tampon, ainsi que le polygone formé par le talus symbolisé. Ce qui ne laisse dans la zone tampon que l'espace vide disponible autour du talus. Si cet espace est trop petit, comme dans la figure 8, l'algorithme ReBankment n'est pas utilisé, et le talus est simplement supprimé de la carte.

Enfin, le cas complexe (4) correspond à des erreurs au moment de la saisie des talus par des opérateurs.

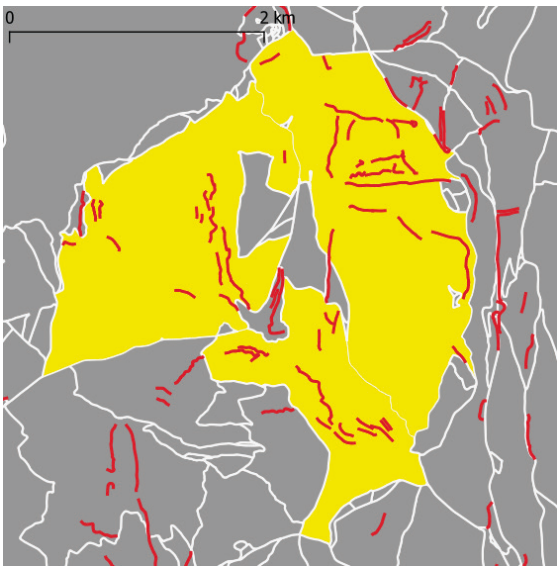

Figure 4 : Trois des plus grandes mailles de notre zone d'étude (en jaune), montrant que le nombre de talus contenus dans ces mailles reste raisonnable pour faire des mailles l'unité de calcul de notre généralisation par moindres carrés.

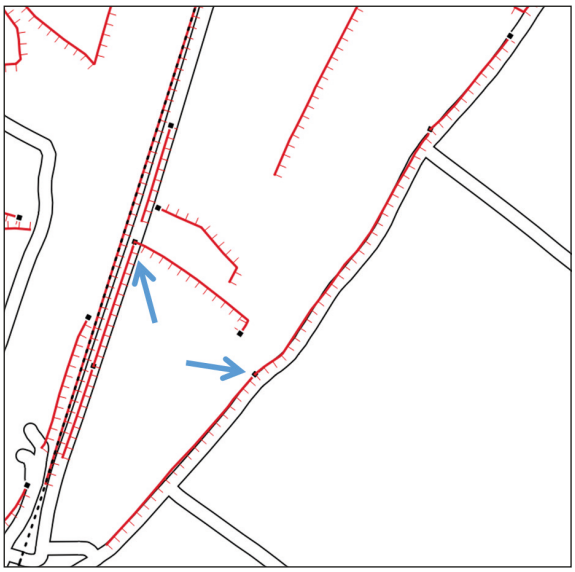

Figure 5 : Certains talus de notre jeu de données sont connectés entre eux, comme si le talus avait été découpé en plusieurs lignes. Pour corriger ce problème, nous réalisons l'opération inverse et nous fusionnons ces talus en une seule ligne avant de la généraliser.

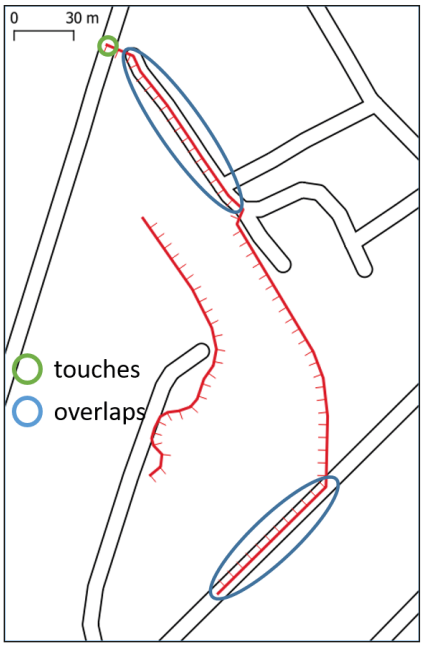

Figure 6 : Exemples de relations spatiales de type « toucher » et « superposer » entre des talus et des routes.

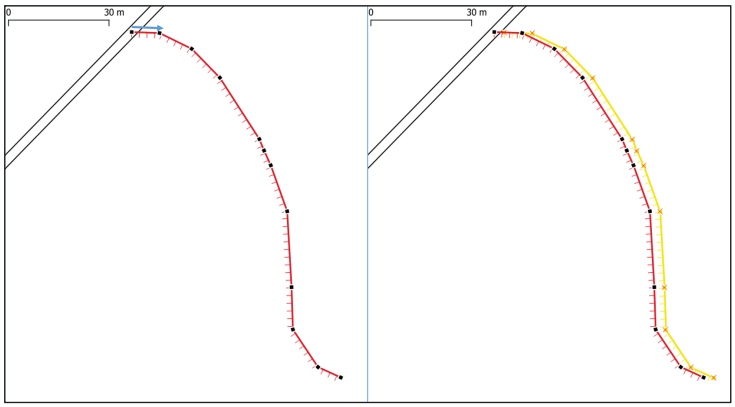

Figure 7 : Exemple de talus touchant une route. A gauche, la flèche bleue montre la translation qui est effectuée pour déplacer très légèrement le talus. A droite, la ligne jaune montre le résultat de cette translation.

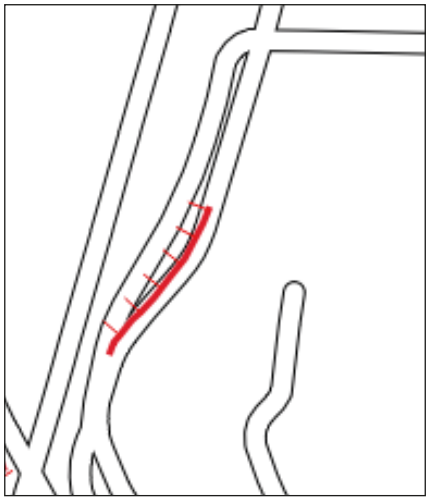

Figure 8 :Certains cas sont trop contraints pour qu'un déplacement valide existe sans déplacer les routes.

Nous avons identifié deux types d'erreurs dans nos données : des talus qui traversent une route ou une rivière (figure 9), et des talus qui s'intersectent entre eux (figure 10). Dans les deux cas, cette situation ne peut se produire sur le terrain, et aucun de ces cas n'est correctement géré par notre algorithme. Dans le cas des talus qui traversent une route, nous décidons de couper le talus en deux au niveau de l'intersection avec la route, puis d'appliquer le pré-traitement qui éloigne les talus qui touchent une route, des deux côtés.

Dans le cas des talus qui s'intersectent (figure 9), nous choisissons de déplacer l'un des talus en employant la même méthode que pour les talus qui « touchent » une route.

Implémentation et résultats

Cas d'étude et implémentation

Notre cas d'étude a les caractéristiques suivantes :

- 55 000 m² dans les Alpes, en France, extraites de la base de données cartographiques utilisée par l'IGN pour produire sa carte au 1:25 000 ;
- 4 258 lignes de talus
- Une zone principalement rurale, avec quelques petites villes.

L'algorithme ReBankment a d'abord été implémenté dans la plateforme de recherche libre sur la généralisation cartographique, CartAGen, en s'appuyant sur les mécanismes de généralisation par moindres carrés déjà présents dans la plateforme (Touya *et al.*, 2019). Ensuite, pour rendre l'algorithme utilisable dans les chaînes de production cartographique de l'IGN, utilisant ArcGIS Pro et PostGIS, nous avons réalisé un portage en Python à l'aide des bibliothèques libres Numpy[1] et Shapely[2]

Résultats

Le traitement de l'ensemble des talus de ce cas d'étude avec l'algorithme ReBankment nous a pris environ 45 minutes sur un PC standard.

Des résultats sont présentés en figure 11. On peut voir sur ces images que les talus sont éloignés correctement des routes, tout en gardant leur forme initiale, même quand cela nécessite une forme de compression de la ligne, comme le talus du bas dans l'image en haut à droite de la figure 11. L'image en bas à droite de la figure 11 montre que ReBankment est également capable de résoudre des conflits de proximité entre deux talus trop proches l'un de l'autre.

Evaluation

Puisque notre algorithme utilise quatre contraintes pour guider sa généralisation, nous utilisons simplement ces quatre contraintes pour évaluer la qualité de nos résultats. Si les contraintes sont satisfaites, c'est que la généralisation est bonne. Nous construisons ainsi des moniteurs de contraintes (Touya et Duchêne, 2011 ; Touya, 2012), c'est-à-dire que nous vérifions la satisfaction de chaque contrainte pour chaque talus dans la carte.

Pour vérifier si (C_2) est satisfaite, nous avons besoin de mesurer la similarité de forme entre le talus avant et après sa généralisation. Ici, les distances usuelles entre deux lignes ne sont pas pertinentes car elles mesureraient surtout le déplacement plutôt que la déformation de la ligne. Donc nous commençons notre mesure de similarité par une translation qui permet de superposer les barycentres des deux lignes. Puis, nous mesurons la similarité en calculant les quatre distances suivantes :

- distance de Hausdorff (m1) ;
- distance de Fréchet discrète (Devogele, 2000) (m2) ;
- Similarité par « time-warping » dynamique (Myers et Rabiner, 1981) (m3)
- distance moyenne (McMaster, 1986) (m4).

Nous utilisons ces quatre mesures parce qu'elles permettent de mettre en valeur des types de similarité différents. (m_1) et (m_2) mesurent l'écart maximum entre les deux lignes, mais (m_2) est plus sensible aux différences de longueur entre les deux lignes. (m_4) mesure l'écart moyen entre les deux lignes, alors que (m_3) mesure la similarité dans la distribution des points dans les deux polylignes. Toutes les valeurs sont exprimées en mètres. Les résultats de cette évaluation sont résumés dans la table 2. Ces résultats montrent que les lignes sont clairement déformées, et ne sont pas uniquement transformées par une translation, ce qui était attendu car une déformation est souvent la meilleure solution dans les zones très contraintes par la largeur des symboles de route. Cependant, cette déformation reste très limitée pour une très grande majorité des talus, avec notamment un tiers des talus qui ne sont déformés que de manière négligeable, et 10 à 15 % seulement des talus qui subissent une déformation importante.

1 https://numpy.org/
2 https://pypi.org/project/Shapely/

Mesure	m_1	m_2	m_3	m_4
Moyenne	31	35	370	22
Médiane	9	9	39	14
% < 5	33	32	15	23
% > 50	14	15	42	10

Tableau 2 : Mesure de similarité de ligne après la translation de la ligne initiale

Les mesures utilisées pour (C_2) sont aussi utiles pour vérifier la satisfaction de (C_1), mais dans ce cas, nous ne faisons pas la translation préalable pour superposer les talus. Les résultats montrent que les lignes sont souvent déplacées : la contrainte n'a pas empêché le déplacement, elle l'a seulement limité. La médiane des valeurs de distance moyenne (McMaster, 1986), par exemple, est de 13,9 m, ce qui représente environ 0,5 mm sur la carte au 1 : 25000. Seulement 2% des talus sont déplacés de manière significative, mais même ce déplacement d'environ 100 m de distance moyenne reste tout à fait raisonnable et nous considérons que la contrainte est satisfaite même pour ces 2%. Pour vérifier si (C_3), la contrainte la plus importante, est satisfaite, nous avons mesuré combien de talus étaient encore situés trop près d'une route après application de l'algorithme. Nous avons trouvé que 80 lignes sur les 3 000 étaient encore trop proches d'une route, ce qui représente 2,5% du nombre de talus. Ce nombre étant relativement conséquent, nous avons essayé de relancer l'algorithme sur ces 80 talus, en augmentant le nombre d'itérations maximum, et cela a permis d'obtenir un bon déplacement pour la plupart d'entre eux. Nous avons gardé un nombre d'itérations relativement petit (100) pour limiter le temps de calcul. Quelques talus nécessitent cependant un plus grand nombre d'itérations pour obtenir un bon déplacement, donc nous pensons qu'un moyen d'améliorer ReBankment serait de l'appliquer en deux passes : (1) on traite d'abord l'ensemble des lignes avec un nombre maximum d'itérations fixé à 100 ; (2) puis on traite les quelques talus restant avec un nombre d'itérations fixé à 500.

Pour la contrainte (C_4), nous n'avons trouvé qu'un seul cas où deux talus étaient trop proches l'un de l'autre après généralisation, sur les 3 000 talus du jeu de données. Là encore, quand les talus sont traités avec un nombre d'itérations maximum à 100, le problème disparaît. Nous considérons donc que cette contrainte est pleinement satisfaite.

Conclusion

En conclusion, cet article a présenté l'algorithme ReBankment, qui déplace des lignes de talus dans une carte topographique au 1 : 25 000, tout en préservant leur forme. Bien que cet algorithme reprenne les principes déjà connus de la généralisation par moindres carrés, la contribution de cette recherche est d'étendre ces principes généraux à la résolution de déplacements élastiques. L'article propose également différentes solutions pour gérer un certain nombre de cas complexes rencontrés dans nos données cartographiques réelles.

Pour aller plus loin, nous pensons tout d'abord que ReBankment pourrait être aussi utilisé pour généraliser des cartes à d'autres échelles, simplement en changeant un peu les poids des contraintes et en augmentant le nombre d'itérations maximum autorisées. Nous aimerions également comparer notre algorithme aux poutres élastiques (Bader *et al.*, 2005) car cet algorithme était utilisé précédemment par l'IGN pour produire ses cartes au 1 : 25 000, mais cela nécessite de réimplémenter cet algorithme, par exemple dans CartAGen (Touya *et al.*, 2019), car il n'existe plus de plateforme où cet algorithme est disponible. Enfin, nous prévoyons d'adapter ReBankment à d'autres problèmes de déplacement élastique de lignes, comme le cas des pistes cyclables le long des routes, ou les fossés (Savino *et al.*, 2011).

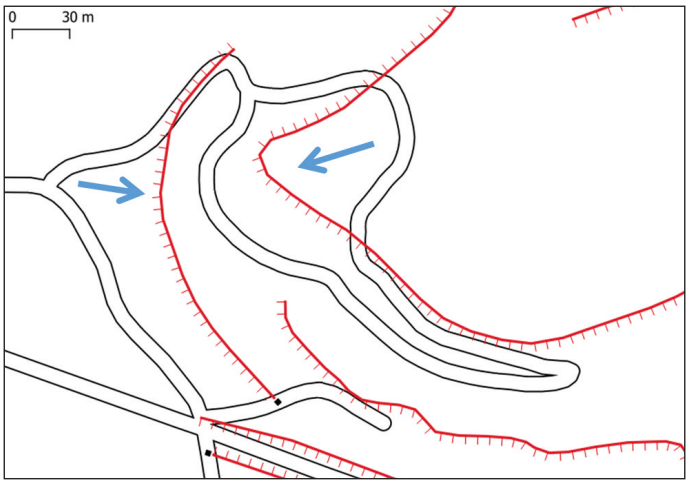

Figure 9 : Deux exemples de talus traversant une route.

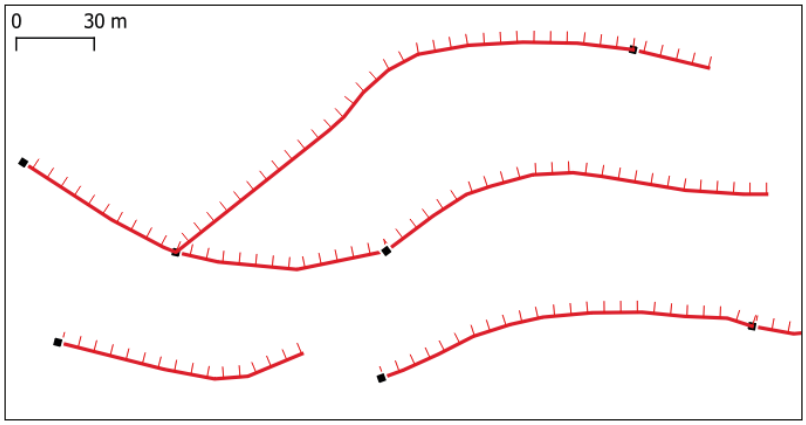

Figure 10 : Exemple de talus qui croise (ou touche) un autre talus en son milieu, ce qui ne peut pas refléter une configuration possible du relief.

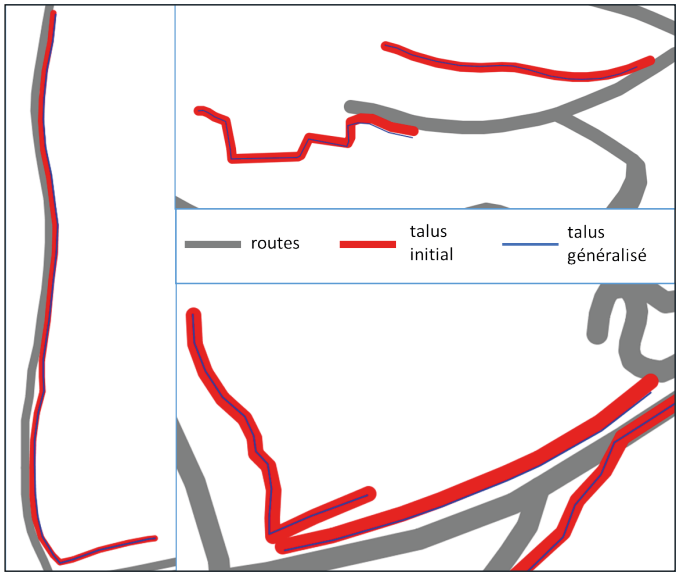

Figure 11 : Trois exemples de déplacement obtenus avec l'algorithme ReBankment. Dans ces images, les symboles ne sont pas ceux utilisés dans la carte pour des raisons de simplicité, mais les largeurs utilisées sont les mêmes pour contraindre le déplacement (les talus initiaux sont les lignes rouges épaisses, les talus généralisés sont les lignes bleues et fines, et les routes en gris).

Bibliographie

Ai T., Xiang Z., Qi Z., Min Y., 2015, « A vector field model to handle the displacement of multiple conflicts in building generalization ». *International Journal of Geographical Information Science,* vol. 29, nº 8, p. 1310–1331. https://doi.org/10.1080/13658816.2015.1019886.

Bader M., Barrault M. (2000) Improving Snakes for Linear Feature Displacement in Cartographic Generalization. Actes du colloque « *Proceedings of the 5th International Conference on GeoComputation* ». Greenwhich, UK. http://www.geocomputation.org/2000/GC034/Gc034.htm.

Bader M., Barrault M., Weibel R., 2005, « Building displacement over a ductile truss ». *International Journal of Geographical Information Science,* vol. 19, nº 8, p. 915–936. https://doi.org/10.1080/13658810500161237.

Brassel K. E., Weibel R., 1988, « A review and conceptual framework of automated map generalization ». *International Journal of Geographical Information Systems,* vol. 2, nº 3, p. 229–244. https://doi.org/10.1080/02693798808927898.

Burghardt D., Meier S. (1997) « Cartographic Displacement using the Snakes Concept ». dans Foerstner W., Pluemer L., *Semantic Modelling for the Acquisition of Topographic Information from Images and Maps*. Basel, Switzerland: Birkhauser.

Dalyot S., Dahinden T., Schulze M. J., *at al.* 2012, « Geometrical Adjustment towards the Alignment of Vector Databases ». *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. I-4, p. 13–18.

Devogele T., 2000, « Mesure d'exactitude et processus de fusion à l'aide de la distance de Fréchet discrète ». *Revue Internationale de Géomatique,* vol. 10, nº 3, p. 359–381.

Dijk, T. C. van, Haunert J.-H., 2014, « Interactive focus maps using least-squares optimization ». *International Journal of Geographical Information Science,* vol. 28, n°10, p. 2052–2075. https://doi.org/10.1080/13658816.2014.887718.

Duchêne C., Ruas A., Cambier C., 2012, « The CartACom model: transforming cartographic features into communicating agents for cartographic generalisation ». *International Journal of Geographical Information Science,* vol. 26, nº 9, p. 1533–1562. https://doi.org/10.1080/13658816.2011.639302.

Egenhofer M. J., Clementini E., Di Felice P., 1994, « Topological relations between regions with holes ». *International Journal of Geographical Information Science,* vol. 8, nº 2, p. 129–142. https://doi.org/10.1080/02693799408901990.

Gaffuri J. (2007) « Field deformation in an agent-based generalisation model: the GAEL model », dans Prbst, F., Kessler, C., *GI-days 2007 - young researches forum*, 30, p. 1–24. IFGI prints. Münster, Germany: IFGI. http://gi-tage.de/archive/2007/downloads/acceptedPapers/gaffuri.pdf.

Gaffuri J. (2009) Three reuse example of a generic deformation model in map generalisation. Actes du colloque « *24th International Cartographic Conference* ». International Cartographic Association. http://icaci.org/documents/ICC_proceedings/ICC2009/html/refer/19_2.pdf.

Gruendig L., Gielsdorf F., Aschoff B. (2007) « Merging Different Data Sets Based on Matching and Adjustment Techniques », dans *Strategic Integration of Surveying Services*. Hong-Kong. http://www.technet-gmbh.de/fileadmin/Geodatenintegration/Publikationen/Paper_FIG_Hongkong.pdf.

Harrie L. E., 1999, « The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization ». *Cartography and Geographic Information Science,* vol. 26, nº 1, p. 55–69. https://doi.org/10.1559/152304099782424884.

Harrie L. E., 2003, « Weight-Setting and Quality Assessment in Simultaneous Graphic Generalization ». *Cartographic Journal, The,* vol. 40, nº 3, p. 221–233. https://doi.org/10.1179/000870403225012925.

Harrie L. E., Sarjakoski T., 2002, « Simultaneous Graphic Generalization of Vector Data Sets ». *Geoinformatica,* vol. 6, nº 3, p. 233–261. https://doi.org/10.1023/a:1019765902987.

Hojholt P., 2000, « Solving Space Conflicts in Map Generalization: Using a Finite Element Method ». *Cartography and Geographic Information Science,* vol. 27, n°1, p. 65–74. https://doi.org/10.1559/152304000783548028.

Kovanen J., Sarjakoski L. T., 2013, « Sequential displacement and grouping of point symbols in a mobile context ». *Journal of Location Based Services,* vol. 7, nº 2, p. 79–97. https://doi.org/10.1080/17489725.2013.764024.

Lecordix F., Le Gallic J.-M., Gondol L. (2007) Development of a new generalisation flow line for topographic maps. Actes du colloque « *Proceedings of 11th ICA Workshop on Generalisation and Multiple Representation* ».

Liu H., Zhang L, Long Y., Zheng Y., 2019, « Real-Time Displacement of Point Symbols Based on Spatial Distribution Characteristics ». *ISPRS International Journal of Geo-Information,* vol. 8, nº 10, 426. https://doi.org/10.3390/ijgi8100426.

Liu Y., Guo Q., Sun Y., Ma X., 2014, « A Combined Approach to Cartographic Displacement for Buildings Based on Skeleton and Improved Elastic Beam Algorithm ». *PLOS ONE,* vol. 9, nº 12, e113953+. https://doi.org/10.1371/journal.pone.0113953.

Lokhat I., Touya G., 2016, « Enhancing building footprints with squaring operations ». *Journal of Spatial Information Science,* vol. 13, p. 33–60. https://doi.org/10.5311/JOSIS.2016.13.276.

Mackaness W. A., 1994, « An Algorithm for Conflict Identification and Feature Displacement in Automated Map Generalization ». *Cartography and Geographic Information Systems,* vol. 21, nº 4, p. 219–232. https://doi.org/10.1559/152304094782540646.

McMaster R. B., 1986, « A Statistical Analysis of Mathematical Measures for Linear Simplification ». *The American Cartographer,* vol. 13, nº 2, p. 103-116. https://doi.org/10.1559/152304086783900059.

Myers C. S., Rabiner L. R., 1981, A comparative study of several dynamic time-warping algorithms for connected-word recognition. *The Bell System Technical Journal,* vol. 60, n°7, p. 1389–1409. https://doi.org/10.1002/bltj.1981.60.issue-7

Nickerson B. G., 1988, « Automated Cartographic Generalization for Linear Features ». *Cartographica,* vol. 25, nº 3, p. 15–66.

Peng D., Haunert J.-H., Wolff A., Hurter C. (2013) Morphing Polylines Based on Least-Squares Adjustment. Actes du colloque « *Proceedings of 16th ICA Workshop on Generalisation and Multiple Representation* ». Dresden, Germany.

Ruas, A., 1998, « A method for building displacement in automated map generalisation ». *International Journal of Geographical Information Science,* vol. 12, n° 8, p. 789–803. https://doi.org/10.1080/136588198241509.

Savino S., Rumor M., Zanon M. (2011) « Pattern Recognition and Typification of Ditches », dans Ruas A., *Advances in Cartography and GIScience*, Lecture Notes in Geoinformation and Cartography, vol. 1, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 425–437. https://doi.org/10.1007/978-3-642-19143-5_24.

Sester M., 2000, « Generalization Based on Least Squares Adjustment ». *International Archives of Photogrammetry and Remote Sensing,* vol. XXXIII, nº B4.

Sester M., 2005, « Optimization approaches for generalization and data abstraction ». *International Journal of Geographical Information Science,* vol. 19, nº 8, p. 871–897. https://doi.org/10.1080/13658810500161179.

Sun Y., Guo Q., Liu Y., Ma X., Weng J., 2016, « An Immune Genetic Algorithm to Buildings Displacement in Cartographic Generalization ». *Transactions in GIS,* vol. 20, nº 4, p. 585–612. https://doi.org/10.1111/tgis.12165.

Touya G. (2010) Relevant Space Partitioning for Collaborative Generalisation. Actes du colloque « *Proceedings of 12th ICA Workshop on Generalisation and Multiple Representation* ». Zurich, Switzerland.

Touya G. (2012) « Social Welfare to Assess the Global Legibility of a Generalized Map », dans Xiao N., Kwan M.-P., Goodchild M. F., Shekhar S., *Geographic Information Science 7th International Conference, GIScience 2012*, Lecture Notes in Computer Science, vol. 7478, Berlin, Heidelberg: Springer Berlin / Heidelberg, p. 198–211. https://doi.org/10.1007/978-3-642-33024-7_15.

Touya G., Berli J, Lokhat I., Regnauld N., 2017, « Experiments to Distribute and Parallelize Map Generalization Processes ». *The Cartographic Journal,* vol. 54, nº 4, p. 322–332. https://doi.org/10.1080/00087041.2017.1413787.

Touya G., Coupé A., Le Jollec J, Dorie O., Fuchs F., 2013, « Conflation Optimized by Least Squares to Maintain Geographic Shapes ». *ISPRS International Journal of Geo-Information,* vol. 2, nº 3, p. 621–644. https://doi.org/10.3390/ijgi2030621.

Touya G., Duchêne C. (2011) « CollaGen: Collaboration between automatic cartographic Generalisation Processes », dans Ruas A., *Advances in Cartography and GIScience*, Lecture Notes in Geoinformation and Cartography, vol. 1, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 541–558. https://doi.org/10.1007/978-3-642-19143-5_30.

Touya G., Lokhat I., Duchêne C. (2019) « CartAGen: An Open Source Research Platform for Map Generalization ». *Proceedings of the ICA,* vol. 2, 134. https://doi.org/10.5194/ica-proc-2-134-2019.

Wei Z., He J., Wang L., Wang Y., Guo Q., 2018, « A Collaborative Displacement Approach for Spatial Conflicts in Urban Building Map Generalization ». *IEEE Access,* vol. 6, 26918–29. https://doi.org/10.1109/ACCESS.2018.2836188.