

# GÉNÉRALISATION DE L'ALGORITHME DE "DOUGLAS ET PEUCKER" POUR DES APPLICATIONS CARTOGRAPHIQUES

Par Xavier BARILLOT, Jean-François HANGOUËT, Hakima KADRI-DAHMANI

Laboratoire Cogit, Institut Géographique National, 2-4 Avenue Pasteur, 94165 Saint-Mandé Cedex, France

Xavier.Barillot@ign.fr, Jean-Francois.Hanguet@ign.fr, Hakima.Kadri-Dahmani@ign.fr

## Résumé

Cette communication reprend les principes mathématiques du fameux algorithme de Douglas et Peucker et montre que s'il a été conçu pour la généralisation du linéaire, il est en fait applicable à des ensembles d'objets de géométrie quelconque.

## Mots clés

Sélection cartographique, généralisation, distance de Hausdorff, Analyse spatiale

## I. Introduction

Dans sa "Théorie de la ligne cartographique", Thomas K. Peucker explique que l'algorithme qu'il a mis au point en collaboration avec David H. Douglas était dédié à la généralisation du linéaire (Peucker 1976 : 138), et qu'à la même époque, bien qu'indépendamment, et pour des applications différentes, il a été décrit dans "deux autres communications au moins". Aujourd'hui, parmi les méthodes de la cartographie vectorielle, l'algorithme de Douglas & Peucker est certainement l'outil de simplification du linéaire le plus populaire, répandu tant dans les SIG du commerce (parce qu'il se programme facilement et qu'il fournit des résultats passables) que dans les articles de recherche (parce qu'il peut être adapté, peaufiné, amélioré de très nombreuses façons). On trouvera récapitulatifs et états de l'art dans (Rimbert 1990), (Cromley-Campbell 1991), (Hershberger-Snoeyink 1992), (Plazanet 1996), (Visvalingam-Herbert 1999).

L'algorithme de Douglas & Peucker prend en entrée une *Polyligne* (une suite ordonnée de points ou, de manière équivalente, de segments), un *Segment* (habituellement les premier et dernier points de la *Polyligne*), et un *Seuil* (un seuil de distance). Par découpage récursif de la *Polyligne*, l'algorithme renvoie une nouvelle suite de segments. Le fonctionnement est le suivant :

$DP(Polyligne, Segment, Seuil)$

$d_{max} \leftarrow$  plus grande distance de la *Polyligne* jusqu'au *Segment*

$p_{max} \leftarrow$  plus lointain sommet de la *Polyligne*

si  $d_{max} > Seuil$  alors

*Polyligne1*  $\leftarrow$  Tous les segments de la *Polyligne* de  $p_i$  (premier sommet) à  $p_{max}$

*Polyligne2*  $\leftarrow$  Tous les segments de la *Polyligne* de  $p_{max}$  à  $p_f$  (dernier sommet)

*Segment 1*  $\leftarrow [p_i, p_{max}]$

*Segment 2*  $\leftarrow [p_{max}, p_f]$

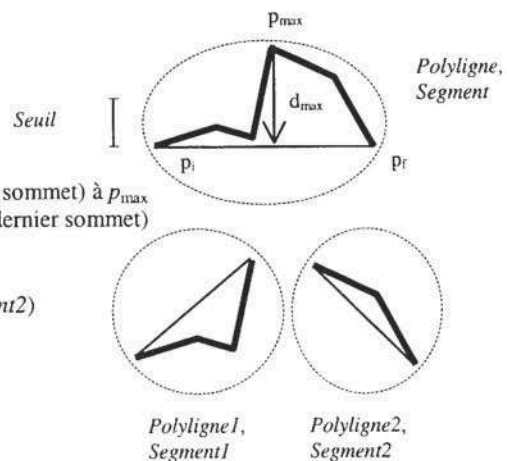
    Renvoie  $DP(Polyligne1, Segment1) \cup DP(Polyligne2, Segment2)$

sinon

    Renvoie {*Segment*}

fin si

fin programme



Douglas et Peucker ont eux-mêmes généralisé leur algorithme à la simplification du relief surfacique, remplaçant la *Polyligne* par le *MNT*, les *segments* par des *triangles*, et la distance maximale des sommets de la ligne

au segment approchant par la distance maximale des sommets du MNT à la facette triangulaire approchante ; l'approximation initiale est constituée par les quatre triangles reliant les coins du MNT au sommet le plus haut (Douglas-Peucker 1975).

Le but de cette communication est de montrer que l'algorithme de Douglas & Peucker, bien que conçu à l'origine pour la généralisation des lignes et des Modèles Numériques de Terrain, peut être utilisé pour la généralisation (soit tel quel abruptement, soit en phase préliminaire) d'objets de toute variété géométrique, au prix (faible) de quelques arrangements mathématiques.

## II. Notations

On trouvera ici la définition des notations que nous utilisons par la suite, sous la forme d'une liste brute qui passe sous silence toute la réflexion qui préside aux choix de notre mathématisation du monde cartographique. Nous espérons que cette mathématisation ne semblera pas trop artificielle à ceux pour qui l'activité n'est pas familière ; qu'elle sera reconnue subtile et potente par ceux qui travaillent sur le sujet : à tous nous ne pouvons promettre ici que de futurs articles sur nos justifications épistémologiques.

### II.1 Généralités

Différence d'ensembles :  $X \setminus Y$  désigne l'ensemble des éléments de l'ensemble  $X$  qui ne sont pas dans l'ensemble  $Y$ .

Ensemble de sous-ensembles :  $\mathcal{P}(X)$  désigne l'ensemble de tous les sous-ensembles (finis) de  $X$ .

### II.2 Géométries et objets

$P$  est le plan, identifié à  $\mathbb{R}^2$ , et mesuré par la distance euclidienne ( $\mathbb{R}$  est l'ensemble des réels).

$p$  est un point du plan :  $p \in P$ .

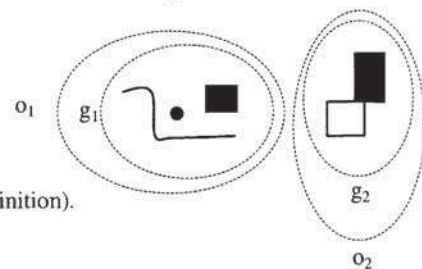
$d(p, q)$  est la distance euclidienne entre les points  $p$  et  $q$ .

$g$  est un ensemble fini et fermé de points du plan.

$o$  est l'ensemble d'un seul  $g$  :  $o = \{g\}$ .

Lapidairement :  $o$  un est objet,

$g$  est une géométrie (non nécessairement connexe, par définition).



On désigne par  $\mathcal{G}$  l'ensemble de tous les  $g$  possibles :  $\mathcal{G}$  est l'ensemble des ensembles fermés bornés de points du plan.

On désigne par  $\mathcal{O}$  l'ensemble de tous les  $o$  possibles :  $\mathcal{O}$  est l'ensemble des ensembles composés d'un ensemble fermé borné de points du plan.

Une *Figure* est un ensemble fini d'éléments de  $\mathcal{O}$ . Une *Figure* appartient à  $\mathcal{P}(\mathcal{O})$  (eg. *Figure* =  $\{o_1, o_2\}$ ).

Pour se donner l'accès à la géométrie d'un objet, on définit l'opérateur  $\gamma$  :

$$\begin{aligned} \gamma: \quad \mathcal{O} &\rightarrow \mathcal{G} \\ o &\mapsto \gamma(o) = g \in \mathcal{G} \mid \{g\} = o \end{aligned}$$

L'accès à la géométrie d'une *Figure* est donné par l'opérateur  $\Gamma$  :

$$\begin{aligned} \Gamma: \quad \mathcal{P}(\mathcal{O}) &\rightarrow \mathcal{G} \\ \text{Figure} &\mapsto \Gamma(\text{Figure}) = \bigcup_{o_i \in \text{Figure}} \gamma(o_i) \end{aligned}$$

### II.3 Mesures de Hausdorff

La fonction suivante  $d_{Hul}$  sera utilisée pour mesurer la plus grande distance spatiale d'un objet vers un ensemble d'objets:

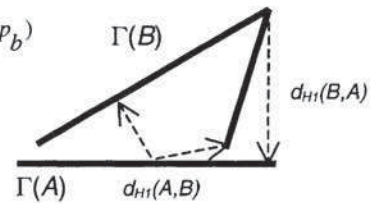
$$\begin{aligned} d_{Hul}: \quad \mathcal{O} \times \mathcal{P}(\mathcal{O}) &\rightarrow \mathbb{R}^+ \\ (o, B) &\mapsto d_{Hul}(o, B) = \max_{p_o \in \gamma(o)} \min_{p_b \in \Gamma(B)} d(p_o, p_b) \end{aligned}$$

La fonction suivante  $d_{H1}$  sera utilisée pour mesurer la plus grande distance spatiale d'un ensemble d'objets (ensemble A) vers un autre ensemble d'objets (ensemble B) :

$$d_{H1} : \mathcal{P}(O) \times \mathcal{P}(O) \rightarrow \mathbb{R}^+ \\ (A, B) \mapsto d_{H1}(A, B) = \max_{p_a \in \Gamma(A)} \min_{p_b \in \Gamma(B)} d(p_a, p_b)$$

Remarque :  $d_{H1}$  n'est pas symétrique.  
Il faut rappeler que  $d_H$  :

$$d_H : \mathcal{P}(O) \times \mathcal{P}(O) \rightarrow \mathbb{R}^+ \\ (A, B) \mapsto d_H(A, B) = \max ( d_{H1}(A, B) , d_{H1}(B, A) )$$



... est une vraie distance mathématique entre des ensembles d'objets identifiés à leur géométrie. On appelle habituellement cette distance la *distance de Hausdorff*. Parce qu'elle exprime l'*éloignement*, elle est souvent utilisée pour l'appariement d'objets dans le domaine du traitement d'image (Galton 1997), et en cartographie automatique pour contrôler les résultats de transformations de généralisation (Hangouët 1995). L'algorithme de Douglas & Peucker l'utilise implicitement : lorsqu'est calculée la plus grande distance de la polygone vers le segment courant, il s'agit bien du calcul de  $d_{H1}(\{\text{Polygone}\}, \{\text{Segment}\})$ , c'est-à-dire de la composante « A vers B » de la distance de Hausdorff...

### III. Généralisation de l'algorithme de Douglas & Peucker

C'est cette dernière observation qui nous a suggéré la généralisation de l'algorithme de Douglas et Peucker que nous proposons ici : la distance de Hausdorff se calcule entre ensembles d'objets de géométrie quelconque, même variée, non obligatoirement entre polygones. En changeant simplement le domaine des paramètres en entrée, en utilisant une méthode générique de découpage des objets, mais en conservant la structure générale de l'algorithme existant, nous disposerons d'un nouvel outil pour construire des « approximations pas trop éloignées » de toute figure donnée. Dans cette partie, nous explicitons tout d'abord la structure générale de l'algorithme, puis nous donnons quelques commentaires importants sur les notions de « découpage » et d'« approximation », ainsi que des possibilités de programmation. Les applications cartographiques proprement dites seront décrites en partie IV.

#### III.1 Douglas & Peucker Généralisé

La structure de notre algorithme est exactement celle de l'algorithme classique de Douglas & Peucker : le premier moment consiste à calculer la plus grande distance de la *Figure* vers l'*Approximation* indiquée ; le second à comparer cette distance à un seuil et à agir en conséquence (en deçà du seuil : accepter l'*Approximation*, au-delà : décomposer la figure en sous-figures, les équiper en approximations individuelles, et ré-appliquer l'algorithme pour chaque couple sous-figure/approximation individuelle). La seule différence fondamentale entre les deux algorithmes réside dans la nature des paramètres : dans notre version, nulle limitation n'est portée à la nature géométrique des composants de la *Figure* (ce n'est plus obligatoirement une polygone seulement) ni à celle des composants de l'*Approximation* (ce n'est plus obligatoirement un segment seulement).

Algorithme DPG (pour : Douglas & Peucker Généralisé) :

Entrée :  $Figure \in \mathcal{P}(O)$ ,  $Proxy \in \mathcal{P}(O)$ ,  $Seuil \in \mathbb{R}^+$   
 Sortie :  $Proxy' \in \mathcal{P}(O)$ ,

DPG( $Figure, Proxy, Seuil$ )

$d_{max} \leftarrow d_{H1}(Figure, Proxy)$  / Composante Hausdorff de  $Figure$  vers  $Proxy$  /

$P_{max} \leftarrow \{ p \in \Gamma(Figure) \mid d_{Hul}(\{p\}, Proxy) = d_{max} \}$  / Points de la  $Figure$  les plus loin de  $Proxy$  /

si  $d_{max} > Seuil$  / Test d'éloignement /

$SubConfigs \leftarrow SPLITS(Figure, Proxy, P_{max})$  / Décomposition de la  $Figure$  et nouvelles approximations /

Renvoie  $\cup_{(Fig_j, Proxy_j) \in SubConfigs} DPG(Fig_j, Proxy_j, Threshold)$  / Récursivité /

sinon

Renvoie  $Proxy$  / Approximation suffisante /

fin si

fin programme

### III.2 Commentaires

#### Découpage

L'algorithme utilise une fonction générique SPLITS ( $Fig, Approx, Pm$ ) dont les caractéristiques doivent être :

Entrée :  $Fig \in \mathcal{P}(O)$ ,  $Approx \in \mathcal{P}(O)$ ,  $Pm \in \mathcal{P}(G)$

Sortie : un ensemble de couples :  $(Fig_i, Approx_i)$  tels que :

$$a/ \quad \cup_i Fig_i = Fig \quad \text{et} \quad \forall i, \forall j, i \neq j, \quad Fig_i \cap Fig_j = \{\}$$

(Avec SPLITS, la figure  $Fig$  est entièrement décomposée en sous-parties, i.e. en ensembles séparés d'objets).

b/ à chaque sous-partie est associée une approximation particulière telle que :

$$\forall i, \quad d_{H1}(Fig_i, Approx_i) < d_{H1}(Fig, Approx)$$

Ceci assure la convergence de DPG, algorithme itératif...

SPLITS consiste à décomposer la  $Figure$  et à composer l' $Approximation$  en fonction des points particuliers donnés par  $Pm$ . La manière particulière de fonctionner de SPLITS, étant donné la  $Figure$ , l' $Approximation$  et quelques points particuliers, est une affaire d'expertise externe, propre aux connaissances et modèles sur les objets manipulés. Dans le cas du Douglas & Peucker habituel,  $Pm$  contient le sommet de la  $Ligne$  (ie.  $Figure$ ) le plus lointain du  $Segment$  (ie.  $Approximation$ ), et SPLITS consiste alors à la fois à couper en deux la ligne en ce point, et à recalculer un segment approchant pour chaque section. Dans les applications de la partie IV, une autre fonction SPLITS est utilisée, dont les caractéristiques d'entrée/sortie sont cependant bien les mêmes.

Démarrage de l'algorithme. Le choix de l' $Approximation$  initiale avec laquelle démarrer l'algorithme est toujours délicat, puisqu'en dépendra, de toute évidence, le résultat final. Le Douglas & Peucker classique démarre en prenant les deux extrémités de la polygone pour segment initial ; lorsque la polygone est fermée (lorsque c'est un polygone), ce sont les deux points de son  $diamètre$  (les deux sommets les plus éloignés l'un de l'autre) qui sont habituellement choisis. La stratégie générale la plus fiable, en ce qui concerne DPG, consiste à commencer avec l'objet ou les quelques objets dont l'extension dans la carte est la plus large. Ce choix est cohérent avec la « philosophie » de l' $éloignement$  qui structure l'interprétation de la distance de Hausdorff.

Fixer le seuil. Le choix du seuil peut sembler être une opération plus évidente, puisque soumise de manière plus directe aux impératifs de l'échelle et du but de la carte : de l'échelle parce que la sélection vise à faire de la place pour la symbolisation aux petites échelles ; du but de la carte parce que la symbolisation elle-même en dépend. Néanmoins, fixer du premier coup le bon seuil (ne parlons même pas de le fixer automatiquement), celui pour lequel l'image produite s'avère harmonieuse à l'échelle et à la symbolisation désirées, reste encore une affaire de chance, et non de déterminisme.

Programmation. Les diagrammes de Voronoï construits sur des points rendent plus rapide la recherche de l'exact éloignement maximal (Krevelde *et al.* 1997) ; construits sur les polygones ils rendent plus rapide le calcul exact de la distance de Hausdorff entre lignes (Alt *et al.* 1991) ainsi qu'entre surfaces (Hangouët 2000). En l'absence de l'outil Voronoï, on peut effectuer des calculs approchés en discrétisant la géométrie des objets : de chaque objet, quelques points sont choisis pour le représenter, et tous les calculs de distance sont effectués sur ces ensembles de points.

#### IV. Applications cartographiques

Pour les diverses applications « cartographiques » données dans cette partie tant pour illustrer l'algorithme que pour préparer la justification de son utilité pour la représentation cartographique, une même fonction SPLITS, simplissime, a été utilisée. Elle est décrite dans le premier paragraphe. On remarque alors que cette méthode de découpage, si on choisit de l'intégrer « en dur » dans l'algorithme général, particularisant celui-ci, conduit à un avatar non-récurif de DPG : la formulation en est donnée dans le deuxième paragraphe. Les trois paragraphes qui s'ensuivent illustrent les résultats de cet algorithme particulier sur trois cas cartographiques : carte de points, carte de lignes, carte d'occupation du sol.

##### IV.1 L'approximation

La fonction SPLITS utilisée ici est la plus simple que nous puissions concevoir dans notre hâte d'effectuer les premiers tests :  $SPLITS_{PRIME}(Fig, Approx, Pm) = \{ ( Fig, Approx \cup \{o_i \in Fig \mid Pm \cap \gamma(o_i) \neq \{\}\} ) \}$ .

Un seul couple  $(Fig, Approx)$  est renvoyé,  $Fig$  étant l'entière figure (pas de découpage en fait !), et  $Approx$  étant simplement augmenté des objets de  $Fig$  dont les géométries contiennent les points indiqués dans  $Pm$ . Cette fonction  $SPLITS_{PRIME}$  a bien les caractéristiques d'une fonction SPLITS (notamment, il est clair qu'ainsi DPG convergera puisque l'approximation prend toujours plus d'éléments de la *Figure* pour se construire, jusqu'à éventuellement les prendre tous).

##### IV.2 Formulation non-réursive

Comme la figure n'est jamais découpée par  $SPLITS_{PRIME}$ , l'intégration de cette fonction « en dur » dans l'algorithme DPG conduit à un algorithme non récurif. La formulation du "Non-Récurif Douglas & Peucker Generalisé" utilisé pour établir les exemples cartographiques donnés plus bas est la suivante :

Entrée :  $Figure \in P(O), \quad Proxy \in P(O), \quad Seuil \in R^+$   
 Sortie :  $Proxy' \in P(O),$

$NRDPG(Figure, Proxy, Seuil)$

$d_{max} \leftarrow d_{H1}(Figure, Proxy)$  / Composante de Hausdorff de *Figure* vers *Proxy* /

tant que  $d_{max} > Seuil$  / Test d'éloignement /

$P_{max} \leftarrow \{ p \in \Gamma(Figure) \mid d_{H1}(\{p\}, Proxy) = d_{max} \}$  / Points de la *Figure* les plus loin de *Proxy* /

$Proxy \leftarrow Proxy \cup \{o_i \in Figure \mid P_{max} \cap \gamma(o_i) \neq \{\}\}$  / *Proxy* augmenté des composants les plus lointains de *Figure* /

$d_{max} \leftarrow d_{H1}(Figure, Proxy)$  / Composante de Hausdorff de *Figure* vers nouveau *Proxy* /

fin tant que

Renvoie *Proxy* / Approximation suffisante /

fin programme

### IV.3 Sur des points

Ici, la *Figure* est composée de points – ça peut être une carte de points, en cartographie. Une bonne approximation initiale, cohérente avec l'idée d'éloignement maximal mise en œuvre par la distance de Hausdorff, est les deux points du *Diamètre maximal* de l'ensemble (ie. les deux points de l'enveloppe convexe les plus éloignés l'un de l'autre). Comme l'illustre la figure ci-dessous, l'algorithme intègre peu à peu chaque point de la *Figure* le plus lointain de l'*Approximation* courante jusqu'à ce que tout nouveau point de la *Figure* tombe « près » d'un point déjà sélectionné, c'est-à-dire dans le rayon spécifié par le seuil.

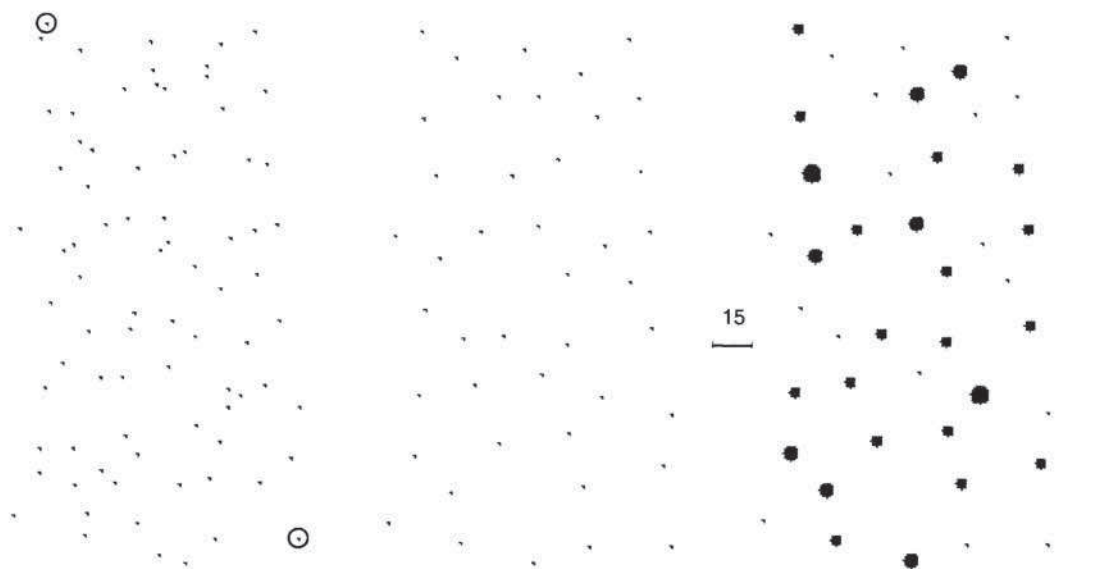


Figure initiale et son Diamètre

NRDPG(Figure,Diamètre, 15)

NRDPG(Figure,Diamètre, 15) +  
Taille des points proportionnelle  
au nombre de voisins alentour  
non sélectionnés.

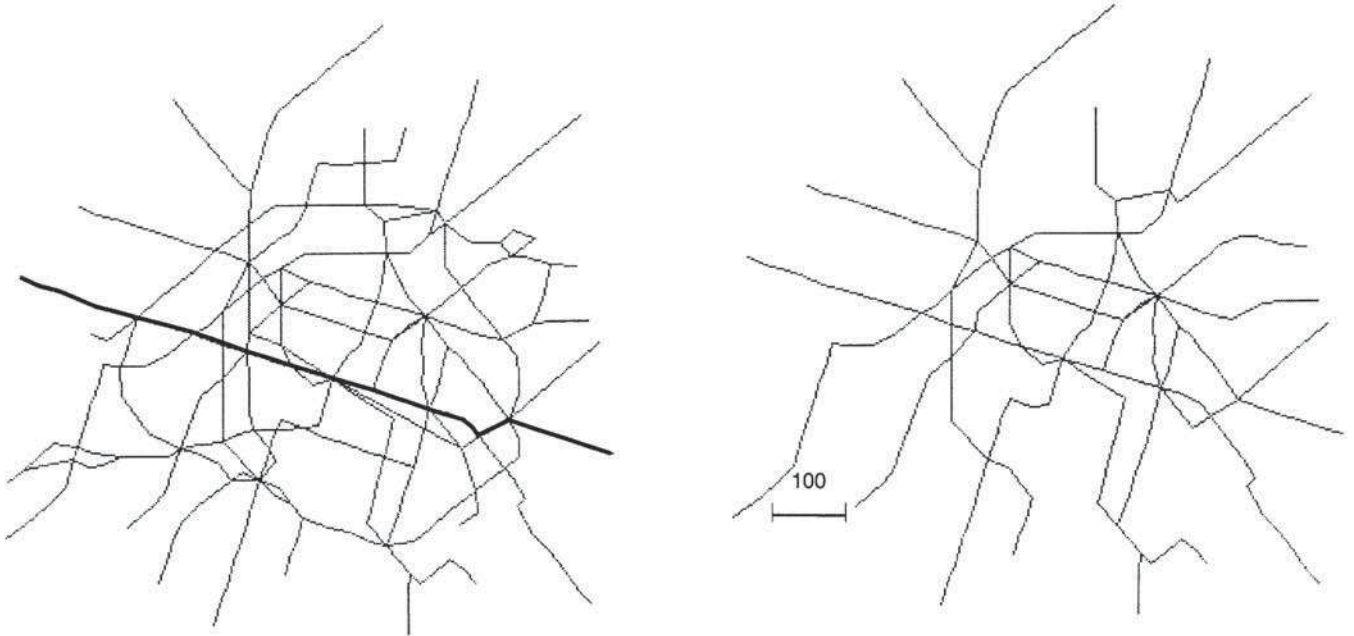
Ceci trouve des applications directes en généralisation des cartes de points aux petites échelles pour lesquelles “*le manque d'espace réclame une sélection intensive*” (Dahlberg 1967:133) et “*une grande part de la spécificité positionnelle inhérente au point (...) est perdue*” (id.:132). Appliquer notre algorithme en choisissant le *Seuil* en fonction de la distance terrain minimale exigée par la lisibilité de la carte à petite échelle conduira à une sélection de points tout à la fois respectueuse des positions et tout à fait lisible. Par ailleurs, comme le montre l'illustration ci-dessus, des « astuces » graphiques peuvent aider à rendre la carte finale encore plus lisible.

Une méthode similaire, quelque peu différente quant au test effectué (1/ commencer avec les points du *Diamètre*, 2/ les mettre dans l'*Approximation*, 3/ trouver le point le plus loin de l'*Approximation*, 4/ répéter l'étape 3/ jusqu'à avoir atteint un certain nombre de points), et dont la parenté avec Douglas et Peucker n'a pas été reconnue, a été mise au point par (Krevelde *et al.* 1997) pour l'affichage automatique de répartitions de villes en fonction de l'échelle à l'écran, avec des résultats visuellement agréables.

### IV.4 Sur des lignes

Ici, la *Figure* est faite de lignes – ce pourrait être un plan de rues (dans lequel on n'a besoin, pour la mathématique de notre algorithme, que de la notion de rue, objet linéaire. La topologie – morceaux de rue, croisements, carrefours etc. – n'intervient pas ; l'Importance – trafic, désignation, longueur, largeur etc. – non plus, même s'il est toujours possible d'inclure des poids dans la fonction distance de Hausdorff utilisée, voire même d'autres fonctions distance). Une bonne approximation initiale, cohérente avec l'idée d'éloignement maximal mise en œuvre par la distance de Hausdorff, est la ligne d'extension maximale, ou l'ensemble de quelques lignes. Dans l'illustration ci-dessous, la

*Figure* est composée des 16 grandes et petites lignes du métro parisien (il ne s'agit que d'une esquisse faite à la souris. Les plans officiels du métro parisien sont infiniment plus beaux).



*Figure* = ensemble des lignes de métro  
Approximation initiale = "Ligne 1"

NRDPG(*Figure*, "Ligne 1", 100)  
Discretisation moyenne: 215 points par ligne de métro

Parce que les deux images se ressemblent, il n'est pas vraiment immédiat de saisir la transformation : on peut dire que les lignes retirées l'ont été de sorte à minimiser la dégradation de la desserte globale de Paris par le métro. On peut comparer le processus au jeu de jonchet (ou mikado), mais avec une contrainte inverse à la stratégie : si au jonchet le joueur commence par retirer les bâtonnets isolés, l'algorithme aura au bout du compte « retiré » les lignes les plus proches d'autres lignes.

L'exemple du métro, s'il illustre la manière de faire de NRDPG, n'est bien sûr guère pertinent en tant que cas de généralisation, puisqu'en utilisateur surtout (moins peut-être en explorateur de la notion de desserte) on s'attend à voir toutes les lignes sur un plan de métro... Un plan de rues, la ville de Montréal par exemple, aurait peut-être donné des résultats d'apparence moins artificielle.

#### IV.5 Sur des surfaces

Ici la *Figure* est composée de zones surfaciques – ce pourrait être une carte thématique d'occupation du sol. Une bonne approximation initiale, pour cette raison désormais familière qu'elle est cohérente avec l'idée d'éloignement maximal mise en œuvre par la distance de Hausdorff, est la zone d'extension maximale, ou un ensemble de quelques zones. Dans l'illustration ci-dessous, la *Figure* est composée de 295 bâtiments (extraits de la BDTopo® de l'IGN sur un village du sud de la France), et l'approximation initiale est constituée des deux bâtiments les plus éloignés. La discrétisation préalable a consisté à prendre des points par bâtiment en nombre proportionnel à sa surface, et équirépartis dans son extension.

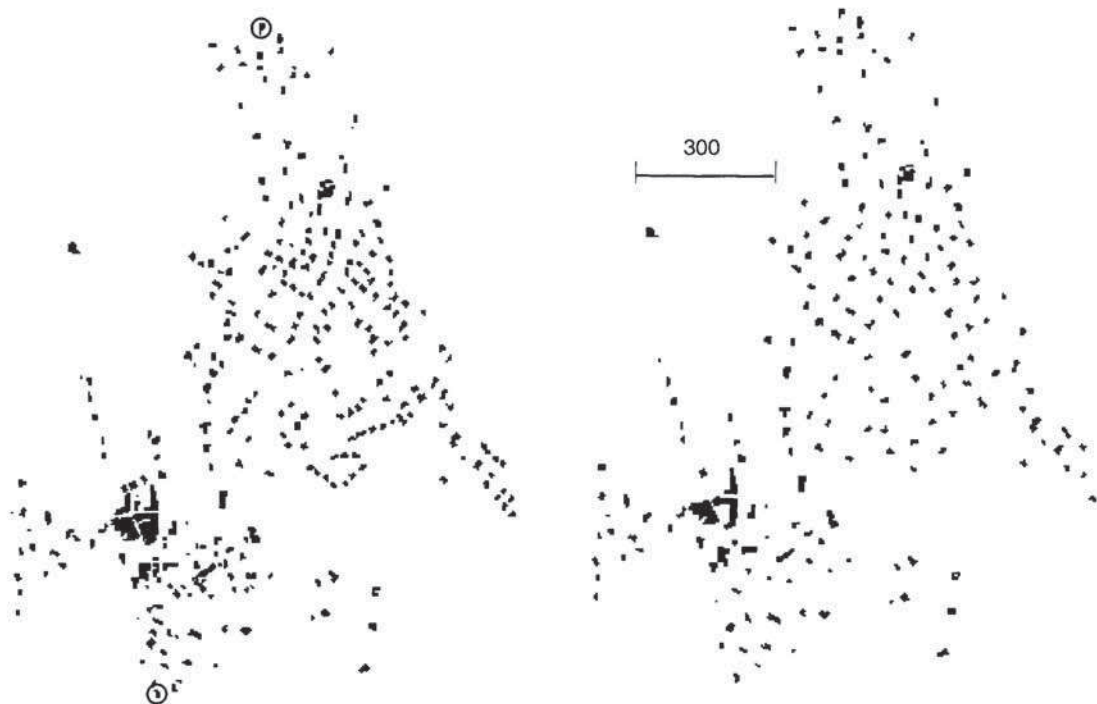


Figure = ensemble de bâtiments (données de la BDTopo®)  
Approximation initiale = bâtiments du *Diamètre*

NRDPG(*Figure, Diamètre, 30*)  
Discrétisation moyenne : 6,7 points par bâtiment

Sur la base de cet exemple, deux critiques principales doivent mettre en garde contre l'utilisation exclusive de NRDPG comme outil de généralisation. Premièrement, l'information micro-structurale n'est pas préservée : les régularités dans les alignements de bâtiments, par exemple, visibles sur la figure originale, ont été perdues. C'est parce que choisir pour l'algorithme un seuil équivalent au diamètre moyen des espaces vides conduit à une distribution homogène du résultat. Ceci dit, il faut remarquer que la perception de l'extension dans l'espace est bien préservée : les limites et la forme du village restent « véridiques ». Deuxièmement, l'algorithme n'a pas conscience de l'importance pratique des bâtiments : il peut négliger des bâtiments dont la taille ou la destination sont remarquables ou importantes pour la carte à fabriquer. Une solution pourrait consister à inclure les bâtiments importants dans l'approximation initiale ; une autre, pour reprendre l'idée émise avec l'application de NRDPG aux réseaux linéaires, serait d'accorder des poids forts aux bâtiments importants pour les éloigner artificiellement « de tout » dans les calculs des distances.

Dans les zones où les bâtiments sont tout à fait similaires, dans lesquelles il n'y a pas de véritables structurations fortes comme les alignements (typiquement, les zones résidentielles sur la Côte d'Azur), NRDPG peut être utilisé tel quel.

## V. Discussion

### V.1 Quelle généralité pour DPG ?

Nous avons généralisé l'algorithme de Douglas et Peucker pour le rendre opérable sur des objets de géométrie quelconque, mais en restant tout de même dans certaines limites, relatives au test effectué et à la distance utilisée.

Test. Le test n'est effectué que sur le critère de *Distance*. Il pourrait aussi être effectué sur la *quantité* d'objets sélectionnés, comparée par exemple au taux de sélection souhaité. Le sens de cette sélection serait tout autre ; il ne s'agirait plus d'une sélection sur l'occupation du territoire et l'extension, elle serait guidée par l'idée de quantité numérique.

Distance. C'est la composante *Figure* vers *Approximation* de la distance de Hausdorff qui est utilisée. On pourrait utiliser la pleine distance de Hausdorff, mais le temps de calcul serait doublé pour de très maigres avantages – comme le démontrent en effet les études statistiques menées par (Abbas-Hottier 1993), l'autre composante, *Approximation* vers *Figure* (de la *Généralisation* vers l'*Original*, dans leur terminologie) est habituellement plus courte que la première, et c'est donc bien celle-ci, la composante *Figure* vers *Approximation*, qui constitue statistiquement la



distance de Hausdorff entre *Figure* et *Approximation*. À la place d'Hausdorff, on peut imaginer d'utiliser toute autre distance mathématique, mais aux dépens peut-être quand même de l'idée d'*éloignement* qui sied si bien au test de distance effectué dans DPG...

## V.2 Domaines ouverts de recherche

Sur des données mixtes. Il nous faudra comparer le résultat de l'algorithme sur une *Figure* composée de bâtiments et de rues, avec les résultats de l'algorithme appliqué séparément aux bâtiments d'un côté et aux rues de l'autre.

Découpage. Pour le moment, les résultats ont été obtenus en utilisant seulement la fonction simpliste SPLITSPRIME. Il nous faudra concevoir et comparer d'autres méthodes spécifiques 1/ de découpage des objets selon leur nature géographique et 2/ de construction des approximations adéquates.

Fixer le seuil. Comment le « meilleur » seuil peut-il être déduit systématiquement de l'échelle et du but de la carte doit encore faire l'objet de recherches tant mathématiques que perceptuelles.

DPG comme outil d'analyse spatiale. Pour une cartographie de qualité, le plein potentiel de DPG ne prétend pas résider dans sa puissance brute de généralisation, mais dans l'aide qu'il peut apporter à l'évaluation non seulement de la présence mais aussi de l'extension de la matière cartographique dans l'espace de la carte, de manière à pouvoir guider des opérateurs de transformation plus locaux ou plus spécifiques. Il s'agit plus d'un outil d'analyse préalable que d'un outil de transformation omnipotent, et les applications d'ailleurs ne sont pas limitées au dessin automatique de carte. Par exemple, pour l'appariement de données géographiques, l'application séparée de DPG sur les différents jeux peut fournir des indications sur l'« apparabilité » des ensembles de représentations (à partir de quel niveau de simplification peuvent-ils être jugés identiques ?). De même, lorsque les deux jeux de données sont de résolution différente, l'application séparée de DPG peut aider à cerner les opérateurs de généralisation (typification, agrégation, sélection régulière,...) impliqués théoriquement dans la transformation idéale qui explique la « marche » entre les deux jeux de données (la différence entre les seuils de DPG qui permettent d'obtenir deux résultats jugés « identiques » est une mesure de la hauteur de la « marche »).

## VI. Conclusion

Dans sa "*Théorie de la ligne cartographique*", Thomas K. Peucker émettait l'espoir que "*d'autres reprissent l'idée et la développassent dans le cadre d'une théorie cartographique toujours plus riche*" (Peucker 1976: 142). En partant non de la théorie de la ligne qu'il proposait, mais des caractéristiques mathématiques les plus intimes de l'algorithme qu'il avait conçu quelques années auparavant, cette communication a pu révéler que les raisons du succès de l'algorithme de Douglas & Peucker résident moins dans la nature spécifique de la *ligne* que dans la nature générale de *toutes choses en configuration*. L'algorithme rend compte de la configuration de manière tout à fait comparable à l'appropriation visuelle : tout d'abord en privilégiant les objets les plus étendus, ensuite dans cette étendue les objets secondaires à nouveau les plus étendus et ainsi de suite... Le fait que les diagrammes de Voronoï ont été rencontrés en chemin, même si ce n'est apparemment que de manière fortuite, au cœur de Hausdorff (paragraphe III.2), n'est certainement pas une coïncidence – mais la formulation voronoïesque d'une pleine théorie spatiale reste à construire...

## References

- Abbas, I., Hottier, P., 1993, *Contrôle du tracé planimétrique d'une carte – Contrôle ponctuel et contrôle linéaire*, Bilan Recherche'92, Bulletin d'Information de l'I.G.N., pp.30-36
- Alt, H., Behrends, B., Blömer, J., 1991, *Approximate Matching of Polygonal Shapes*, Proceedings of the 7<sup>th</sup> ACM Symposium on Computational Geometry, pp.186-193
- Cromley, R., Campbell, G., 1991, *Noninferior Bandwidth Line Simplification: Algorithm and Structural Analysis*, Geographical Analysis, Vol.23, No.1 (January 1991), pp.25-38
- Dahlberg, R., 1967, *Towards the Improvement of the Dot Map*, Annuaire International de Cartographie, vol.VII, pp.158-167
- Douglas, D., Peucker, T., 1973, *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature*, The Canadian Cartographer, Vol.10, No.2, December 1973, pp.112-122
- Douglas, D., Peucker, T., 1975, *Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data*, Computer Graphics, Visions and Image Processing, Vol.4, No.2, December 1973, pp.375-387
- Galton, A., 1997, *Continuous Change in Spatial Regions, in Spatial Information Theory*, Proceedings of Cosit'97, Laurel Highlands, Pennsylvania, USA, October 1997. Hirtle & Frank (Eds.) Springer, Lecture Notes in Computer Science #1329, pp.1-13
- Hangouët, JF., 1995, *Computation of the Hausdorff Distance Between Two PlaneVector Polygones*, Proceedings of AutoCarto 12, Charlotte N.C., pp. 1-10
- Hangouët, JF., 2000, *Storing Voronoi Diagrams in Geographical Databases*, Proceedings of GeoComputation 2000 (CDROM ISBN :0-9533477-2-9) University of Greenwich, Chatham Maritime, U.K., August 23-25
- Hershberger, J., Snoeyink, J., 1992, *Speeding Up the Douglas-Peucker Line-Simplification Algorithm*, Proceedings of the 5<sup>th</sup> International Symposium on Spatial Data Handling, Charleston, South Carolina, vol.1, pp.134-143
- van Kreveld, M., van Oostrum, R., Snoeyink, J., 1997, *Efficient settlement selection for interactive display*. Proceedings Auto-Carto 13: ACSM/ASPRS Annual Convention Technical Papers, 1997
- Peucker, T., 1976, *A Theory of the Cartographic Line*, Annuaire International de Cartographie, vol.XVI, pp.134-143

- Plazanet, C., 1996, *Enrichissement des bases de données géographiques: analyse de la géométrie des objets linéaires pour la généralisation cartographique (application aux routes)*, Thèse de doctorat, Université de Marne-la-Vallée
- Rimbert, S., 1990, *Carto-graphies*, éditions Hermès
- Visvalingam, M., Herbert, S., 1999, *A Computer Science Perspective on the Bendsimplification Algorithm*, *Cartography and Geographic Information Science*, Vol.26, No. 4, pp.253-270