Distance discrète de Fréchet optimisée

Thomas Devogele, Maxence Esnault, Laurent Etienne

Distance discrète de Fréchet optimisée

Devogele Thomas[1], Esnault Maxence[1], Etienne Laurent[1]

1. Laboratoire d'informatique de l'Université François Rabelais
64 avenue Jean Portalis
37200 Tours France
prenom.nom@univ-tours.fr

RESUME. Le calcul de distance entre des polylignes est essentiel pour mesurer la similarité entre les géométries de deux objets linéaires. Cet article décrit une nouvelle version de la distance de Fréchet discrète. Cette version optimise grandement le temps de calcul de la distance et améliore la précision du résultat. Parallèlement, elle définit des appariements de meilleure qualité pour les extrémités des segments des deux polylignes. Ces propriétés sont indispensables pour manipuler de gros volumes de polylignes ayant des nombres d'extrémités de segments importants.

ABSTRACT. Distance computation between polylines is a key point to assess similarity between geometrical objects. This paper describes a new optimized algorithm to compute discrete Fréchet distance which lower computation time and improve precision. Moreover, this algorithm defines a matching path between polylines points. Thanks to this algorithm, big data polyline repositories can be mined.

MOTS-CLES : Distance de Fréchet, distance linéaire, appariement, mesure de similarité

KEYWORDS: Fréchet distance, linear distance, data matching process, similarity measure

1. Introduction

Les Systèmes d'Information Géographique (SIG) disposent de fonctionnalités de comparaison d'objets géométriques. Ces fonctionnalités s'appuient sur différents types de mesures de similarité. Parmi ces dernières les mesures de distances entre deux géométries tiennent une place prépondérante. Pour la comparaison de points, des mesures de distances exactes existent. Ces mesures varient en fonction du contexte et de différents systèmes de référence : distance euclidienne, distance de Manhattan, distance d'Haversine, etc. En ce qui concerne les objets linéaires, le choix de la mesure approprié est plus complexe. Ces mesures répondent à des objectifs différents. Le calcul de la distance minimale est indispensable pour sélectionner les points de deux lignes les plus proches afin de les relier. Par exemple, la longueur minimale d'un pont entre deux rives d'une rivière peut ainsi être définie. Des distances moyennes ou maximales sont aussi fort utiles pour mesurer l'écart moyen ou maximal entre deux lignes. Ces distances s'apparentent plus aux notions de similarité entre deux objets. Elles sont employées pour calculer les écarts entre des points similaires de deux lignes. Elles peuvent également être employées pour contrôler la qualité d'une ligne en la comparent à une ligne plus détaillée (Devogele 2000). Ces distances sont aussi utilisées pour regrouper les objets de deux bases de données géographiques représentant les mêmes phénomènes du monde réel (Alt et al. 2001, Mustière et Devogele2008) voir les fusionner à l'aide processus de déformations élastiques (Doytsher et al 2001, Devogele 2002). Finalement, elles servent à mesure la similitude entre des objets linéaires afin par exemple de les regrouper dans des clusters (Etienne et al. 2016).

La difficulté majeure de ces mesures de similarité porte sur le processus d'appariement. Il consiste à extraite les couples de points homologues entre deux lignes. En effet l'écart entre les deux lignes est déduit des distances ponctuelles entre leurs points homologues. De plus, d'autres problèmes se posent dans le choix de la mesure de similarité entre deux lignes :

- Les mesures doivent être précises.
- Le temps de calcul doit être raisonnable afin de pouvoir mesurer les écarts d'un grand nombre de lignes composées d'un nombre élevé de points.

Finalement, ces mesures doivent disposer de certaines propriétés. Par exemple, elles doivent pouvoir être employées sur des lignes ouvertes ou fermées, être robustes et pouvoir être utilisées pour comparer de grand nombre de lignes de longueurs différentes. Finalement, les résultats obtenus doivent être invariants au sens de parcours des deux lignes.

Dans, cet article, nous nous focaliserons sur la recherche d'une distance linéaire maximale générique entre deux polylignes. Une polyligne est une suite de segments de droites reliés à leurs extrémités. Un bref état de l'art sur les distances les plus usitées sera fait dans le chapitre 2. Puis nous proposerons une distance de Fréchet optimisé dans le chapitre 3. L'optimisation porte à la fois sur le temps de calcul et sur la précision du résultat.

2. Etat de l'art

Pour calculer la similarité entre deux polylignes, trois mesures sont fréquemment employées : La distance d'Hausdorff, la distance de Fréchet et le Dynamic Time Warping (DTW). Dans ce chapitre, elles vont être présentées. Une synthèse conclut cette partie.

2.1. Distance d'Hausdorff

La distance d'Hausdorff est employée pour rendre compte de l'écart maximum entre deux polylignes (L_1, L_2) (Taha et Hanbury 2015). Par définition, deux polylignes L_1 et L_2 sont à une distance de Hausdorff (d_H) l'une de l'autre inférieure à *d* unités, si chaque point de L_1 est à moins de *d* unités d'au moins un point de L_2, et si, réciproquement, chaque point de L_2 est distant de moins de *d* unités d'au moins un point de L_1.

La distance de Hausdorff est définie (équation 1) comme la plus grande des deux composantes suivantes :

- d1 qui est la plus grande valeur de la distance non symétrique de L1 à L2,
- d2 qui est la plus grande valeur de la distance non symétrique de L2 à L1.

$$d_1 = \max_{p_1 \in L_1} \left[\min_{p_2 \in L_2} \left[dist(p_1, p_2) \right] \right]$$
$$d_2 = \max_{p_2 \in L_2} \left[\min_{p_1 \in L_{11}} \left[dist(p_2, p_1) \right] \right] \quad (1)$$
$$d_H = \max(d_1, d_2)$$

La figure 1 illustre le calcul des deux composantes de la distance de Hausdorff.

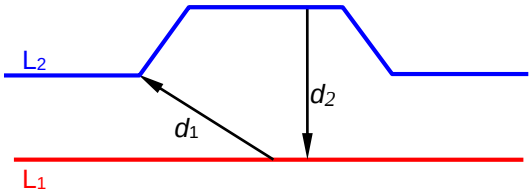

Figure 1 : Exemple de mesure de la distance de Hausdorff

Cette distance a l'avantage de fournir deux mesures. Des lignes d'emprise différente peuvent ainsi être comparées à l'aide de la composante partant de la plus petite ligne. Par contre, la distance de Hausdorff, à l'inconvénient de calculer la distance sur les couples de points les plus proches et non sur des points homologues. Les points homologues sont les points qui visuellement se correspondent. Par exemple, pour la figure 1, le point de L_2 servant à calculer d_1, n'est pas le correspondant intuitif du point de L_1, c'est simplement le point le plus proche. La distance de Hausdorff considère les polylignes comme de simples ensembles de points non ordonnés. Ce problème est particulièrement important pour les lignes très sinueuses ou avec des boucles. Des distances faibles peuvent alors être renvoyées pour

des lignes dissimilaires. De même, les couples de points ne peuvent pas être considérés comme des appariements. Néanmoins, cette particularité à l'avantage de réduire le temps de calcul : la complexité algorithmique de cet algorithme est linéaire (Taha et Hanbury 2015).

2.2. Distance de Fréchet

La distance de Fréchet est une distance tenant compte de l'orientation des lignes. Elle s'appuie sur la propriété suivante : Toute polyligne orientée est équivalente à une application continue f : [a, b] $\rightarrow$ V ou a, b $\in \Re$, a < b et V est l'espace vectoriel. La distance de Fréchet (d_F) est la suivante :

Soit f : [a, a'] $\rightarrow$ V et g :[b, b'] $\rightarrow$ V deux polylignes et || || la norme usuelle

$$d_F(f,g) = \inf_{\substack{\alpha:[0,1]\rightarrow[a,a'] \\ \beta:[0,1]\rightarrow[b,b']}} \max_{t\in[0,1]} \left\| f\left(\left(\alpha(t)\right)\right) - g\left(\beta(t)\right)\right\| \quad (2)$$

Une illustration intuitive de la distance de Fréchet est la suivante : un maître et son chien suivent deux chemins. Ils avancent ou s'arrêtent à volonté, indépendamment l'un de l'autre, mais ils ne peuvent pas revenir sur leurs pas. La distance de Fréchet entre ces deux chemins est la longueur minimale de la laisse qui permet de réaliser un cheminement de concert satisfaisant ces conditions.

La distance de Fréchet (d_F) a l'avantage d'être calculée uniquement à partir de couples de points homologues. Hélas, cette distance a l'inconvénient d'être complexe à programmer. Un algorithme d'ordre O(N M $\log^2$(N M)) avec N et M le nombre d'extrémités des segments des polylignes est donné dans (Alt et Godau 1992). Dans (Eiter et Mannila 1994) une version discrète est proposée. Elle adopte une approche par programmation dynamique en deux étapes. Dans un premier temps une matrice des distances (MD), de taille NM, entre les points de L_1 et les points de L_2 est calculée.

$$MDi,j = d\,(L1.i,L2.j) \quad (3)$$

Dans un deuxième temps, une matrice ; appelée matrice de Fréchet (MF), est définie. La valeur de la cellule en i,j est calculée de la manière suivante :

$$MFi,j = \max\,(MDi,j,\ \min(MFi\text{-}1,j\ ,\ MFi,j\text{-}1\ ,\ MFi\text{-}1,j\text{-}1)). \quad (4)$$

La distance de Fréchet discrète est la valeur de $MF_{n,m}$.

Cet algorithme est d'ordre O(NM), mais le résultat obtenu est approché. Néanmoins, l'erreur est bornée par la longueur du plus grand segment des deux lignes (Eiter et Mannila 1994). En ce qui concerne l'appariement, dans (Devogele 2002), l'ensemble des chemins possibles avec une laisse de longueur égale à d_F sont envisagés. Celui dont la distance moyenne entre les couples de points est la plus faible est retenu. Chaque couple correspond alors à des points appariés. L'évaluation de l'ensemble des chemins possible est cependant couteuse. Une amélioration du temps de calcul est proposée par (Etienne et al. 2016) en n'évaluant pas l'ensemble des chemins, mais en le calculant dynamiquement par backtracking. Elle consiste à partir du bas à gauche de la matrice MF et à remonter en choisissant une des trois cellules possibles $MF_{i-1,j}$, $MF_{i,j-1}$ et $MF_{i-1,j-1}$. La cellule ayant la valeur la plus faible est retenue. En cas d'égalité, la valeur de la cellule de MD au même indice est employée.

La valeur la plus faible est retenue. Cette approche est plus rapide, mais l'appariement est parfois de moins bonne qualité pour des lignes complexes. En effet seul des critères locaux sont pris en compte. Or pour ce type de lignes, une optimisation locale n'entraine pas une optimisation globale. De même, l'appariement n'est pas identique lorsque le sens de parcours des deux lignes est inversé. La figure 2 donne des exemples d'appariements (segments en pointillés) obtenus pour deux lignes (segments reliés par des extrémités bleues ou rouges) en fonction du sens de parcours (normal puis inverse). Les lignes qui servent d'exemple ont pour coordonnées {(0.2,2) ; (1.5,2.8) ; (2.3,1.6) ; (2.9,1.8) ; (4.1,3.1) ; (5.6,2.9) ; (7.2,1.3) ; (8.2,1.1)} et { (0.3,1.6); (3.2,3.0) ; (3.8,1.8) ; (5.2,3.1) ; (6.5,2.8) ; (7,0.8) ; (8.9,0.6)}. La valeur distance discrète obtenue (1,69) est la même et est portée par le même couple de points (le deuxième à gauche avec l'étiquette MAX). Néanmoins l'appariement est différent pour deux points. Un appariement entre deux points n'apparaît pas dans le parcours sens inverse. Le deuxième résultat est visuellement meilleur.

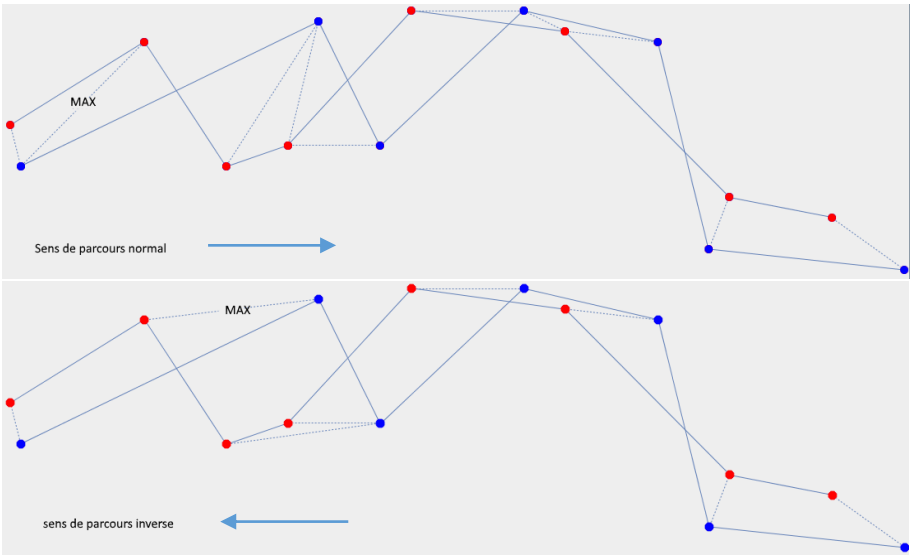

Figure 2 : Appariements à l'aide de la distance de Fréchet discrète

2.3. Dynamic Time Warping (DTW)

La similarité calculée par l'algorithme de Dynamic Time Warping (déformation temporelle dynamique) (Berndt et Clifford 1994) permet de mesurer la somme des déformations nécessaires pour passer d'une ligne à une autre. L'algorithme DTW est très employé, par exemple en traitement du signal. Comme l'algorithme de calcul de la distance de Fréchet discrète, il a une complexité d'ordre O(NM) et il se calcule par programmation dynamique. Cet algorithme cherche à minimiser la somme des écarts entre les deux lignes. Les valeurs des cellules de la matrice (DTW) se calculent de la manière suivante :

$$DTW_{i,j} = MD_{i,j} + \min(DTW_{i-1,j} , DTW_{i,j-1} , DTW_{i-1,j-1})). \quad (5)$$

Le résultat de l'algorithme est obtenu par lecture de la valeur de $DTW_{n,m}$. Cet algorithme renvoie une mesure de similarité très intéressante, mais n'est pas une distance. Elle ne vérifie pas l'inégalité triangulaire. Cette mesure ne peut pas être employée pour calculer des similarités entre des ensembles de lignes de longueurs différentes. Ce défaut est problématique pour mesurer la similarité entre des couples de lignes de longueurs différentes. DTW surestime la similarité entre des lignes de petite taille vis-à-vis de lignes plus longues. De même, elle ne permet pas facilement de manipuler des lignes fermées. Le DTW a été développé pour des séries temporelles de même durée, cela explique la difficulté à l'adapter à des lignes de tailles différentes ou fermées. En ce qui concerne le processus d'appariement, DTW renvoie des couples de bonne qualité visuelle. De plus, l'appariement est invariant au sens de parcours des deux lignes. En revanche, cet algorithme peut choisir d'apparier des points parfois très éloignés si ce choix permet de minimiser la somme des autres distances ponctuelles. La figure 3 présente le résultat de l'appariement. La mesure du DTW est la somme des longueurs qui relient les points appariés. La valeur MAX des longueurs est 1,712. Elle est plus grande que celle du la DF : 1,697 et est porté par un autre couple de points. Visuellement, les appariements sont bons. Pour cet exemple, l'appariement visuellement est meilleur, les virages du bas sont appariés avec des virages du bas et les virages du haut avec des virages du haut.

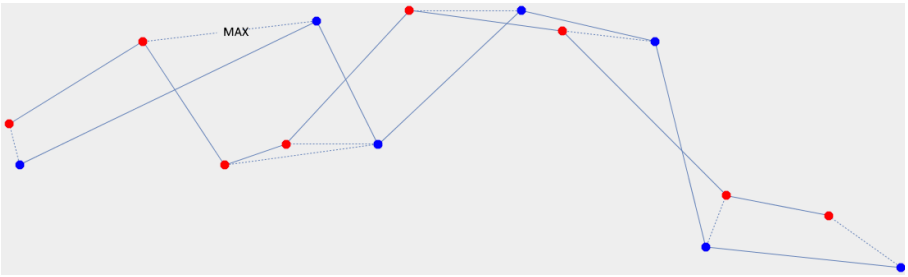

Figure 3 : Appariement à l'aide de l'algorithme de DTW

TABLE 1. Synthèse des trois mesures de similarités

	Ordre des points	Temps de calcul	Appariement	approximation	Robustesse et généricité	Appariement Invariant par inversion
Hausdorff	Non	linéaire	+	Non	Non	Oui
Fréchet discrète	Oui	N M	++	Oui	Oui	Non
DTW	Oui	N M	+++	Ce n'est pas une distance	Non	Oui

2.4. Synthèse

Le tableau 1 compare ces trois mesures. La distance de Hausdorff qui ne considère pas les lignes comme des ensembles ordonnés et le DTW qui ne permet pas de gérer des ensembles de lignes de taille différente ne répondent pas à nos critères. Pour la distance de Fréchet, un algorithme optimisé doit être proposé pour renvoyer un résultat précis avec un ordre de complexité plus simple.

3. Distance de Fréchet discrète optimisée

Dans ce chapitre, une distance de Fréchet discrète optimisée est proposée. L'objectif de ce nouvel algorithme est de diminuer les temps de calcul, fournir un résultat plus précis et invariant en cas d'inversion de parcours des deux lignes. Pour illustrer cet algorithme, les deux polylignes de l'état de l'art sont utilisées comme exemple.

3.1. Optimisation du calcul de la matrice de distance (MD)

La complexité O(NM) de l'algorithme de calcul de la distance de Fréchet discrète est liée au calcul de l'ensemble des cellules des deux matrices. Les distances de Fréchet sont généralement utilisées pour comparer des lignes proches qui se ressemblent. En effet, pour constater que deux lignes ne se ressemblent pas, des méthodes beaucoup plus rapides existent. Elles se basent essentiellement sur des rectangles englobants ou des bandes Epsilons (Gabay et Doytsher1994). C'est pourquoi, pour des polylignes similaires, on considère que le chemin retenu a de très fortes chances d'être proche de la diagonale. L'algorithme proposé est constitué de quatre étapes. Dans un premier temps, les valeurs de la diagonale de la matrice MD sont calculées. Pour les matrices presque carrées, une fois le bord de la matrice atteint, nous continuons les calculs en incrémentant uniquement l'indice de ligne ou de colonne restant. Pour les autres matrices rectangles, une fonction permettant d'approximer le numéro de colonne en fonction du numéro de ligne doit être employée. Cette fonction permet ainsi de suivre au mieux la diagonale. Dans un deuxième temps, la valeur maximale calculée sur cette diagonale : DiagMax, est conservée comme paramètre des deux étapes suivantes. Cette diagonale peut être vue comme un premier chemin possible. La valeur DiagMax pour ce chemin est la longueur provisoire de la laisse (distance de Fréchet discrète).

Dans la troisième étape de l'algorithme, la partie en bas à gauche de la diagonale de MD est calculée partiellement. Pour chaque colonne j, la valeur en dessous de la diagonale ($MD_{j+1,j}$) est calculée et stockée si sa valeur est inférieure à DiagMax. Les cellules en dessous sont testées avec le même prédicat. Un deuxième test d'arrêt est ajouté : l'indice de la ligne doit être supérieur à celui de la colonne précédente.

Dans la quatrième étape, la partie en haut à droite de la diagonale de MD est calculée partiellement. Pour chaque ligne i, la valeur à droite de la diagonale ($MD_{i,i+1}$) est calculée et stockée si sa valeur est inférieure à DiagMax. Les cellules à droite sont testées avec le même prédicat. Un deuxième test d'arrêt est ajouté : l'indice de la colonne doit être supérieur à celui de la ligne précédente. En effet, il n'est pas nécessaire de calculer des cellules qui ne seront pas utilisées par un chemin optimal sachant qu'un meilleur chemin (la diagonale) est déjà connu. L'algorithme 1 décrit ces 4 étapes pour une matrice presque carré. La figure 4 fournit le calcul partiel de la matrice MD. La distance euclidienne entre deux points a été utilisée. Elle peut être remplacée par une autre distance en fonction du contexte. Pour cet exemple, les valeurs des cellules (en jaune) de la diagonale sont calculées. La valeur DiagMax vaut 2,642 (en gras sur la figure 4).

MD

L1

num	x	y
0	0,2	2
1	1,5	2,8
2	2,3	1,6
3	2,9	1,8
4	4,1	3,1
5	5,6	2,9
6	7,2	1,3
7	8,2	1,1

L2

num	0	1	2	3	4	5	6
X	0,3	3,2	3,8	5,2	6,5	7	8,9
Y	1,6	3	1,8	3,1	2,8	0,8	0,6

0,412	3,162	3,606	5,120	6,351	6,905	8,812
1,697	1,712	2,508	3,712	5,000	5,852	7,720
2,000	1,664	1,513	3,265	4,368	4,768	6,675
2,608	1,237	0,900	**2,642**	3,736	4,220	6,119
4,085	0,906	1,334	1,100	2,419	3,701	5,412
5,457	2,402	2,110	0,447	0,906	2,524	4,022
6,907	4,346	3,437	2,691	1,655	0,539	1,838
7,916	5,349	4,455	3,606	2,404	1,237	0,860

Figure 4 : Matrice MD de l'exemple

Algorithme 1 : Calcul Optimisé de la matrice de distance (MD)

```
// Calcul de la diagonale et de DiagMax
i = 0 ; j=0 ; DiagMax = 0 ;
Tant que i<(NbreLigne-1) et j<(NbreColonne-1)
        MD[i][j] =  d(L1.i,L2.j)
        Si (MD[i][j]> DiagMax) alors DiagMax = MD[i][j] ;
        Si (i<NbreLigne-1) alors i++
        Si (j<NbreColonne-1) alors j++
fin tant que
// Calcul de la partie haut-droit de MD
jprecedant = 0 ;
Pour (i=0 ; i<NbreLigne; i++)
        j=i;
        Tant que (j<NbreColonne ET DiagMax > d(L1.i,L2.j))
                        OU (j<NbreColonne ET j<jprecedant)
                MD[i][j] = d(L1.i,L2.j);
                j++ ;
        Fin Tant que
        jprecedant = j ;
Fin Pour
//Calcul de la partie bas-gauche de MD
iprecedant = 0 ;
Pour (j=0 ; j<NbreLigne;j++)
        i=j
        Tant que (i<NbreLigne ET max > distance_euclidienne(i,j))
                        OU (i<NbreLigne ET i<iprecedant)
                MD[i][j] =d(L1.i,L2.j);
                i++ ;
        Fin Tant que
        iprecedant = i ;
Fin Pour
```

Puis les deux parties (bas gauche et haut droite) de la matrice sont calculées partiellement. Les cellules blanches sont calculées et stockées. Les cellules grisées foncées sont calculées mais ne sont pas stockées. Les cellules grisées claires ne sont pas calculées. Leur valeur est laissée écrite à titre indicatif. Par exemple, pour la colonne 0, les cellule $MD_{1,0}$, $MD_{2,0}$ et $MD_{3,0}$ sont calculées et stockées, la valeur de la cellule $MD_{4,0}$ n'est pas stockée. Sa valeur calculée (4,085) est supérieure à DiagMax

(2,642). Pour cet exemple, seules 17 cellules sont stockées pour la partie basse gauche. Une seule cellule de la partie haute droite est stockée. Seuls 46 % des valeurs des $MD_{i,j}$ sont stockées. Plus la matrice est grande plus ce pourcentage de cellules stockées est faible pour des polylignes proches.

3.2. Optimisation du calcul de la matrice de Fréchet

L'étape de calcul de la matrice de Fréchet est similaire à l'algorithme classique. Par contre, seules les cellules ayant les mêmes indices que les cellules de MD prises en compte sont calculées avec la formule suivante :

$$MF_{i,j} = Max\begin{pmatrix} MD_{i,j} \\ Min\begin{pmatrix} MF_{i-1,j}\ si\ MD_{i-1,j}\ est\ stockée \\ MF_{i,j-1}\ si\ MD_{i,j-1}\ est\ stockée \\ MF_{i-1,j-1}\ si\ MD_{i-1,j-1}\ est\ stockée \end{pmatrix} \end{pmatrix} \quad (6)$$

L'équation 6 est une version modifiée de l'équation 4. Elle ne prend en compte uniquement les cellules stockées. Il n'est pas nécessaire de calculer les autres valeurs, sachant que le chemin optimal ne passe pas par ces cellules. La figure 5, fournit la matrice MF de l'exemple.

		L2	num	0	1	2	3	4	5	6
			X	0,3	3,2	3,8	5,2	6,5	7	8,9
L1			Y	1,6	3	1,8	3,1	2,8	0,8	0,6
num	x	y								
0	0,2	2		0,412						
1	1,5	2,8		1,697	1,712	2,508				
2	2,3	1,6		2,000	1,697	1,697				
3	2,9	1,8		2,608	1,697	1,697	2,642			
4	4,1	3,1			1,697	1,697	1,697	2,419		
5	5,6	2,9			2,402	2,110	1,697	1,697	2,524	
6	7,2	1,3						1,697	1,697	1,838
7	8,2	1,1						2,404	1,697	**1,697**

Figure 5 : Matrice de Fréchet (MF) de l'exemple (appariement en rouge)

Le résultat obtenu par cette méthode optimisée est similaire au calcul classique de la distance de Fréchet discrète. Pour cet exemple, la distance de Fréchet discrète (dF_d) vaut 1,697 (valeur écrite en rouge dans la matrice MD de la figure 4). Cette valeur est surestimée et l'erreur maximale est toujours bornée par la longueur du plus long segment des deux lignes (Eiter et Mannila 1994). Les étapes suivantes de l'algorithme visent à améliorer la précision de cette distance. Elle nécessite dans un premier temps d'améliorer le processus d'appariement.

3.3 Ajout des points perpendiculaires

L'algorithme de calcul de la distance de Fréchet discrète fournit un résultat approché, car il ne s'appuie que sur les extrémités des segments. Pour obtenir un résultat exact, il faut prendre en compte d'autres points situés sur les segments. Le

nombre de points des segments étant potentiellement infini, la propriété suivante est employée : la distance de Fréchet entre deux polylignes est portée par un couple de points dont au moins l'un des deux est une extrémité de segment. L'autre point est soit le projeté orthogonal de l'extrémité sur un segment soit une extrémité de segment de l'autre polyligne. La figure 6 illustre cette propriété sur un cas simple. Pour ces deux lignes, la distance entre leurs points augmente. Elle atteint son maximum pour le couple entre $L_{1.1}$ et son projeté orthogonal puis elle diminue.

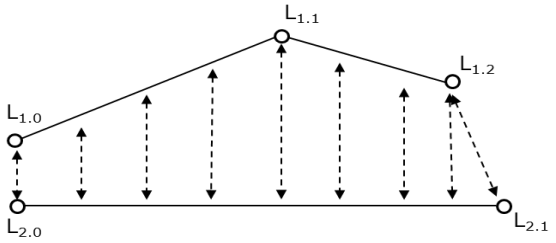

Figure 6 : évolution de la distance entre les points des polylignes

Algorithme 2 Ajout des projeté orthogonaux dans les deux lignes

Supprimer de MD les valeurs supérieures à dF_d (Distance de Fréchet discrète)
// Pour ajouter des points à L_2
Pour chaque point $L_{1.i}$ de L_1
  Pour chaque couple de points $L_{2.j}$, $L_{2.j+1}$
    Si les deux cellules [i,j] et [i,j+1] sont stockées dans MD
      Si le projeté P de $L_{1.i}$ sur le segment [$L_{2.j}$ $L_{2.j+1}$] est un nouveau point
        alors ajouter P à L_2
    Si une seule des deux cellules est stockée
      Si le projeté P de $L_{1.i}$ sur le segment [$L_{2.j}$ $L_{2.j+1}$] est un nouveau point
      et si d($L_{1.i}$,P)< dF_d
        alors ajouter P à L_2
// Pour ajouter des points à L1
Pour chaque point $L_{2.j}$ de L_2
  Pour chaque couple de points $L_{1.i}$, $L_{1.i+1}$
    Si les deux cellules [i,j] et [i+1,j] sont stockées
      Si le projeté P de $L_{2.j}$ sur le segment ($L_{1.i}$ $L_{1.i+1}$) est un nouveau point
        alors ajouter P à L_1
    Si une seule des deux cellules est stockée
      Si le projeté P de $L_{2.j}$ sur le segment [$L_{1.i}$ $L_{1.i+1}$] est un nouveau point
      et si d(P, L2.j)< dF_d
        alors ajouter P à L1

Pour chaque polyligne, des points correspondant aux projetés orthogonaux doivent donc être ajoutés aux deux lignes avant de mesurer la distance de manière plus précise. Afin de réduire le temps de calcul et le nombre de points à ajouter, pour chaque extrémité $L_{1.i}$, seuls les segments [$L_{2.j}$, $L_{2.j+1}$] dont [i,j] ou [i,j+1] est stockée dans MD, sont pris en compte. L'opération identique est effectuée pour les extrémités $L_{2.j}$. En effet, seuls de nouveaux points sur ces segments peuvent porter potentiellement la distance de Fréchet (dF). De même, afin de réduire le nombre de cellule à prendre en

compte, les valeurs stockées supérieures à la valeur approximée de dF : 1,697 sont supprimées de MD. Pour notre exemple, 11 valeurs sont retirées. L'algorithme 2 résume ce processus d'enrichissement par l'ajout des projetés orthogonaux.

Sur la figure 7, les points issus de la projection orthogonale sont ajoutés aux deux lignes. Ils sont représentés par des carrées. Le nombre de points de chaque ligne augmente (double ou triple en moyenne). Pour l'exemple, la ligne 1 enrichie est composée des 8 extrémités (rond rouge) de la ligne 1 et de 5 points projetés (carré rouge). De même la ligne 2 enrichie est composée des 7 extrémités de la ligne 2 (rond bleu) et de 12 points projetés (carré bleu). Cette solution est préférable à un sur-échantillonnage classique (Devogele 2000) qui divise chaque segment afin que leur longueur soit inférieure à un seuil. En effet cette dernière n'assure pas pour autant que les points optimaux soient ajoutés. Empiriquement, cette méthode donne de très bons résultats. Néanmoins, il reste à démontrer que l'ajout des projetés orthogonaux, renvoie bien la distance de Fréchet exacte pour des polylignes.

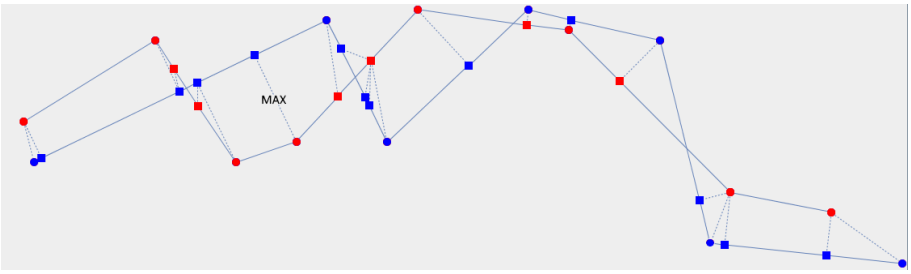

Figure 7 : lignes enrichies des points projetés, et appariement entre les points.

3.4. Calcul des deux matrices pour les lignes enrichies

Une fois les lignes L_1 et L_2 enrichies des points ajoutés pour obtenir L_{1+} et L_{2+}, les deux premières étapes : calcul de MD et de MF (voir chapitre 3.1 et 3.2) peuvent être reprises. La seule variante est le changement du seuil. La valeur DiagMax n'a pas besoin d'être calculée et elle est remplacée par la distance dF_d. En effet, le parcours du chemin optimal nécessite une laisse de longueur inférieure ou égale à celle du parcours de L_1 et L_2. Pour l'exemple, la distance de Fréchet dF obtenue est de 0,950. Cette distance est bien inférieure à la distance de Fréchet discrète (dF_d = 1,697). Elle est portée par les points reliés par la ligne en pointillé estampillé MAX sur la figure 7.

3.5. Amélioration de l'appariement

Parallèlement, à l'optimisation et à l'amélioration de la précision de la mesure, le processus d'appariement défini par backtraking doit être modifié. Il est rapide, mais ne il souffre de deux problèmes :

- Il n'est pas invariant au sens de parcours des polylignes,
- Des couples d'appariements peuvent être inutiles.

Pour résoudre ces problèmes, une modification est proposée, elle consiste à diviser la matrice de Fréchet en quatre parties (voir figure 8) :

- A) la partie haute gauche allant du couple de points portant dF et remontant vers les premiers points,
- B) la partie basse droite allant du couple de points portant dF et descendant vers les derniers points,
- C & D) les deux autres parties ne sont pas considérées. Le chemin optimal ne passe pas par ces cellules.

Pour la partie haute gauche (A), le backtraking s'effectue sur la MF des lignes parcourue en sens normal. Pour la partie basse droite (B), le backtraking s'effectue sur la MF des lignes parcourue en sens inverse. L'utilisation de ces deux processus de backtraking, permet de réduire de manière significative le nombre de chemins alternatifs possibles. De plus, par construction cet algorithme fournit un appariement invariant au sens du parcours.

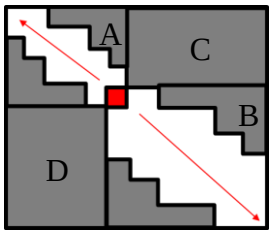

Figure 8 : décomposition de la matrice de Fréchet (MF)

Pour illustrer cette nouvelle méthode, l'appariement des extrémités de L_1 et L_2 est maintenant détaillé. Pour L_1 et L_2, c'est le couple de points $L_{1.1}$ et $L_{2.0}$ qui porte la distance de Fréchet discrète (cellule [1,0] en rouge sur la matrice de la figure 4). La partie haute gauche est donc réduite à deux cellules [0,0] et [1,0] et le backtraking est trivial. Pour la partie basse droite, elle est limitée par les cellules [1,0] et [7,6]. Le backtraking utilise la matrice de Fréchet partiel du parcours inverse des lignes (voir figure 9).

L2 inverse	num	6	5	4	3	2	1	0
	X	0,3	3,2	3,8	5,2	6,5	7	8,9
L1 inverse	Y	1,6	3	1,8	3,1	2,8	0,8	0,6

num	x	y							
7	0,2	2	0,860	1,237					
6	1,5	2,8		0,860	1,655				
5	2,3	1,6			0,906	0,906			
4	2,9	1,8				1,100	1,334	1,334	
3	4,1	3,1					1,100	1,237	
2	5,6	2,9					1,513	1,664	
1	7,2	1,3							1,697
0	8,2	1,1							**1,697**

Figure 9 : Partie haute gauche de la MF (lignes parcourus en sens inverse)

La ligne du point $L_{1.0}$ n'est pas considérée. Le backtraking est simple. Le seul test d'égalité entre des valeurs de MF qui renvoie vrai, concerne le choix entre les cellules

[2,3] et [2,4] en venant de [3,3]. Elles ont pour valeur 0,906. Le couple [2,3] est choisi, car MD[2,3] < MD[2,4]. Le chemin décrivant les appariements est défini par les cellules où la valeur est écrite en rouge. Pour cet exemple simple, les appariements sont similaires à ceux du parcours des lignes sens inverse (figure 2 bas). En comparant les matrices de la figure 5 et de la figure 9, nous constatons que le nombre de chemins possibles est bien plus faible pour la figure 9. Les valeurs rencontrées par le processus de backtraking sont de plus en plus faible. Par contre pour la figure 5, ces valeurs se maintiennent à la distance de Fréchet tant que le couple portant cette distance n'est pas rencontré ce qui oblige à considérer plusieurs alternatives. Pour l'appariement des points de L_{1+} et de L_{2+}, les traits en pointillés de la figure 7, relient les points appariés.

3.6. Processus et complexité

Ce nouveau processus enchaîne 6 étapes (voir figure 10).

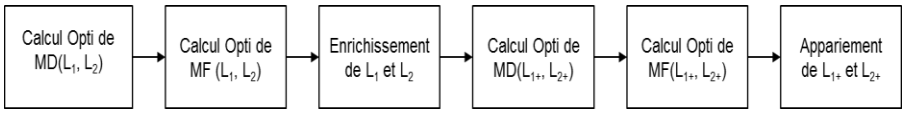

Figure 10 : Etapes du processus de mesure de la dF et de l'appariement

Il est donc plus compliqué que l'algorithme classique relativement simple. Cependant, du point de vue de la complexité algorithmique, il est optimisé. La complexité au pire ne change pas. Elle est toujours d'ordre O(NM) pour des lignes dissimilaires. Par contre, la complexité en moyenne pour des polylignes proches, est beaucoup plus faible. En effet, pour les étapes de calcul des matrices optimisé $MD(L_1,L_2)$ et $MF(L_1,L_2)$, seules les valeurs proches de la diagonale sont prises en compte. Cela équivaut à ne considérer que la diagonale et les cellules incluses dans une bande de largeur variable autour de cette diagonale. Si la largeur moyenne de la bande est estimée à k nous pouvons estimer l'ordre de la complexité à k fois la longueur de la diagonale ($\sqrt{N^2+M^2}$). Pour des lignes ayant un grand nombre de points, k étant une constante petite par rapport à N et M, la complexité en moyenne est d'ordre O($\sqrt{N^2+M^2}$). Il faut noter que des optimisations similaires sont proposées pour le calcul du DTW pour des séries temporelles. Par exemple (Rakthanmanon et al. 2012) utilise une bande de largeur fixe autour de la diagonale. Dans ce cas, la largeur de la bande doit être définie par l'utilisateur. Pour l'algorithme de distance de Fréchet discrète optimisée, il n'y a pas de paramètre et la largeur de la bande varie automatiquement. La complexité de l'étape d'enrichissement consiste à calculer pour tous les points de L_1 et L_2, k' projetés orthogonaux. La constante k' varie, mais reste petite, la complexité est donc de l'ordre O(N+M). Finalement, la complexité du calcul des matrices optimisées $MD(L_{1+},L_{2+})$ et $MF(L_{1+},L_{2+})$ dépend aussi du nombre de points de L_{1+} et de L_{2+}. Ces nombres étant défini par k', N et M. La complexité en moyenne reste d'ordre O($\sqrt{N^2+M^2}$). La dernière étape : l'appariement par backtraking, se déroulant le long de la diagonale, sa complexité est aussi du même ordre que les deux étapes précédentes. Pour résumer, la complexité de cet algorithme est donc d'ordre O($\sqrt{N^2+M^2}$). Chaque étape ayant une complexité de cet ordre ou inférieure.

4. Conclusion

Un algorithme optimisé de mesure de la distance de Fréchet discrète a été présenté dans cet article. Le temps de calcul est réduit de manière significative passant de O(NM) à O($\sqrt{N^2 + M^2}$). Cette réduction est indispensable pour traiter les gros volumes de lignes qui peuvent maintenant être obtenues à l'aide de capteur GPS ou des VGI (Volunteered geographic information). La phase d'enrichissement améliore aussi la précision de la mesure. Cependant, il reste à démontrer que le résultat obtenu est égal à la distance de Fréchet (non discrète) pour des polylignes. Finalement, un processus d'appariement plus performant lui est associé. Cet algorithme doit maintenant être validé sur des jeux de données volumineux pour mesurer les gains obtenus en termes de temps de calcul et de précision. De même, des processus de parallélisation ou de filtrage vont être aussi testés pour diminuer les temps de calcul.

Références

Alt, H., & Godau, M. (1992, July). Measuring the resemblance of polygonal curves. In *Proceedings of the eighth annual symposium on Computational geometry* (pp. 102-109). ACM.

Alt, H., Knauer, C., & Wenk, C. (2001). Matching polygonal curves with respect to the Fréchet distance. In *STACS 2001* (pp. 63-74). Springer Berlin Heidelberg.

Berndt D, Clifford J (1994) Using dynamic time warping to find patterns in time series. AAAI-94 workshop on knowledge discovery in databases, pp 229–248

Devogele, T. (2000). Mesure d'exactitude et processus de fusion à l'aide de la distance de Fréchet discrète. *Revue internationale de Géomatique*, *10*(3-4), 359-381.

Devogele, T. (2002). A new Merging process for data integration based on the discrete Fréchet distance. In *Advances in spatial data handling* (pp. 167-181). Springer Berlin Heidelberg.

Doytsher, Y., Filin, S., & Ezra, E. (2001). Transformation of datasets in a linear-based map conflation framework. *Surveying and Land Information Systems*, *61*(3), 165-176.

Eiter, T., & Mannila, H. (1994). *Computing discrete Fréchet distance*. Tech. Report CD-TR 94/64, Information Systems Department, Technical University of Vienna.

Etienne, L., Devogele, T., Buckin, M., Mcardle, G. (2016) Trajectory Box Plot: a new pattern to summarize movements, International Journal of Geographical Information Science (IJGIS), Taylor & Francis, Analysis of Movement Data, vol. 30, num. 5, pp.835-853

Y. Gabay and Y. Doytsher (1994) Automatic adjustment of line maps, GIS/LIS, pp 233-241.

Mustière, S., & Devogele, T. (2008). Matching networks with different levels of detail. *GeoInformatica*, *12*(4), 435-453.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., ... & Keogh, E. (2012). Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD* (pp. 262-270).

Taha, A. A., & Hanbury, A. (2015). An efficient algorithm for calculating the exact Hausdorff distance. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 37(11), 2153-2163.