A New Merging Process for Data Integration Based on the Discrete Fréchet Distance

Thomas Devogele

Naval academy Research Institute (IRENav), Lanvéoc, BP 600, F-29 240 Brest Naval, FRANCE, devogele@ecole-navale.fr

Abstract

The overlay process is currently one of the main computational solutions used to integrate several data layers from different sources. Unfortunately, it is problematic when trying to overlay many layers. This leads to several geometric problems such as the management of sliver polygons. This paper proposes a new merging process to complement the vector overlay for data integration of several layers. This process, based on measures derived from the Fréchet distance, matches common points (either lines or polygons). It also merges an ordered set of pairs of matching points (vertices) into a single geometry.
Keywords: data integration, overlay, merging, data matching, Fréchet distance

1 Introduction

Overlaying geospatial data is a frequently required and computationally complex procedure in a Geographic Information System (GIS). The overlay process was one of the first ways through which users combined map information to produce new maps.

Most GISs use map layers to structure geographical objects and each layer describes some particular aspect of the real world. Organisation of data into several map layers is a well-known technique, which often strives to achieve efficiencies in data storage and manipulation. However, these structures can be computationally costly when data from different layers, or when several layers of data must be combined in efforts to resolve a spatial query. Often the data layers can be distributed among remote databases and must be identified and accessed using solutions such as those proposed by the NSDI (USGS 1998).

Currently, four kinds of constraints limit a successful application of the overlay process:

- **Projections and coordinate systems** can be different. To combine two data layers, the projection and coordinate system of one of these data sets must often be converted. This conversion process can be a source of error.
- **Accuracy and scale** are generally difficult aspects with which to deal. Although each layer may be reasonably accurate within the scale limitations, differences in input errors between the two layers will most likely cause mismatches between the two geospatial references. Moreover, the scales and levels of generalisation of these data sets can be incongruent. Some other conflicts (e.g. semantics, resolution) between the object representations complicate the overlay process (Parent *et al.* 1996)
- **Limited arithmetic precision** of computers.
- **Homologous objects**. Some objects can be represented in both data sets. If the two layers are combined, the objects may be redundant in the resulting data set. However, integration of these redundant objects is necessary to control the resulting layer accuracy (Chrisman 2001) and to integrate complementary a-spatial descriptions of some phenomena (Devogele et al. 1998). By extension, homologous geometries are defined as geometries of homologous objects. In the same way, homologous points or vertices are defined as points that represent the same part of an object. (e.g. points that represent the same turn of a road).

Therefore, to ensure an efficient overlay of layers, some semantic and computational constraints must be defined. First, layers must have the same scale and projection. Secondly, the object geometry should not be replicated in two different layers. As data can come from different sources, however, these conditions cannot always be guaranteed. Therefore, several manual pre-processes and post-processes must be implemented to adjust the overlay result. These processes are unwieldy, time consuming and error prone.. Therefore, identification of a generic and automated overlaying process is, as yet, an unresolved objective for future GIS research.

This paper proposes a new merging operation that can be defined as an automatic pre-process that improves the overlay results of either lines or polygons. It is based on a data matching process and measures derived from the Fréchet distance. The reminder of this paper is organised as follows. Section 2 briefly outlines the basics of the overlay process, fuzzy tolerances and merge process. Section 3 describes data matching between homologous geometries (either lines or polygons) and the discrete Fréchet distance used by the merging process. Section 4 introduces the new merging process based on the matching process and a weighting function**.** Finally Section 5 concludes our paper and outlines some further work.

2 Overlay Process

The overlay process can handle different types of data layers (e.g. tiles, network, Digital Terrain Model). So far many overlay processes have been defined (e.g. CAD-type overlay, Boolean overlay, rules overlay) (cf. (Chrisman 2001) for a survey). Overlay processes depend on the type of geometry (e.g. polygon, line) and the type of expected result. Whatever the kind of overlay process, the input is made of two or more layers issued from the same or different sources, and the output is a new layer in which the new geometries (e.g. points, line) are defined as a function of the input geometry.

2.1 Errors and Corrections

Constraints that limit a successful application of the overlay process create **sliver polygons** when a basic overlay process is applied (see Fig. 1c). These polygons are small artefacts created during an overlay and result from slight differences in the geometric presentation of boundaries that should have been the same.

To resolve these errors, a fuzzy tolerance (Fig. 1d) can be associated with the overlay process (Dougenik 1980). The fuzzy tolerance is a distance where intersections and points are treated as coincident. In other words, it is the smallest distance between two geometries. If the distance between two points (vertices or segment point) is less than the fuzzy tolerance, these two points are merged. The fuzzy tolerance also resolves dangling lines and dangling nodes that should *logically* be connected at two ends. Dangling lines are dead-end lines that are connected to other lines at one end only. The other end is a dangling node.

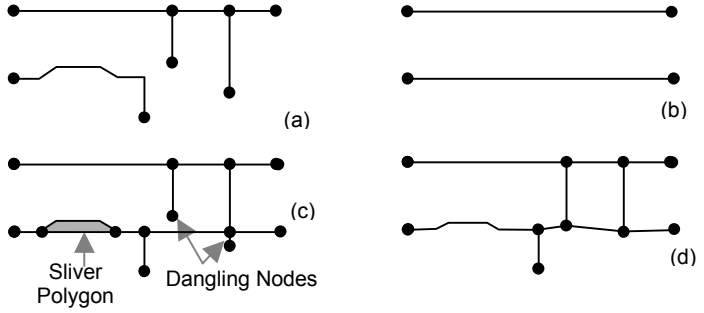

Fig. 1. Overlay examples; **(a)** and **(b)** input layers, **(c)** output without fuzzy tolerance, **(d)** output with fuzzy tolerance

One challenge is to distinguish between small geometric differences of homologous objects from different objects that are close or nearby. Either the distance is set too small and many errors are not removed, or it is set too large and some distinct points are merged (for example, a real dead-end close to a road). Moreover, the overlay with fuzzy tolerance leads to new inaccuracies. For

example, new positional errors can "creep" into a GIS (Pullar 1993). (Harvey and Vauglin 1996) improves this process by introducing multiple tolerances. The setting of the tolerance is fixed according to the application purpose, scale and on statistical analysis of line form and error.

Fuzzy tolerance is a method used to merge geometry locally. Unfortunately, this method does not take into account the 'global' geometry. Indeed, different objects, which are close, cannot be distinguished from homologous objects, due to measures between their geometries. Similarly, the fuzzy tolerance process merges closest points (e.g. using Euclidean distance (d_E)). However, the two points that may be merged are not necessarily the homologous points (Alt and Godau 1995).

2.2 Merging Process for Homologous Geometry

For homologous geometry, the data integration process must use a better merging process than the one of fuzzy tolerance. The input of this process is two geometries and the output is a single geometry. Few merging processes are defined due to the fact that is very difficult to distinguish homologous geometries from closed geometries and to match the points of a common geometry.

Some processes have been proposed like a pre-process of Adjust (TCI 1999). Adjust is a rubber sheeting process. Rubber sheeting transformation is non-linearly distorted to force-fit some geographic elements (e.g. points, lines) to specific positions. As implemented in Adjust, a semi-automatic process assembles AutoCAD layers. A user selects two sets of points (e.g. crossroads in network layers) and then correlates them using From-To relationships. It also takes some lines, which should logically coincide. The process automatically interpolates any number of matching points (so-called calibration pairs). These matching points are intermediate points on these lines. Two modes are available to automate the setting of calibration pairs: "vertex to vertex" or "divide distance" (see Fig. 2).

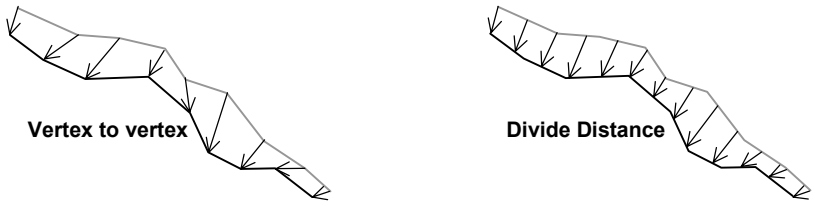

Fig. 2. Two modes of automate matching points of Adjust

These matching points are merged when the rubber sheeting are used. The "From" point and the "To" point are merged into the "To" point.

We can remark that:

- The "From-To" is a one-to-one relationship. A point cannot relate to many points. Section 3 will show that these relationships can be one-to-many or many-to-one.

- The output point is the “To” point. This solution is adequate if the layer of the point “To” is much more accurate than the "From" one. If the **accuracy** (relationship between a measurement and the reality represented) of the layers are equivalent, a "middle" point should be a better solution.
- Calibration pairs take into account the order of the vertices or points of lines. In other words, for the vertex to vertex mode, for two oriented line L_1 and L_2, the calibration pair following ($L_{1.i}$, $L_{2.j}$) are ($L_{1.i+1}$, $L_{2.j+1}$) with $L_{1.i}$ $L_{1.i+1}$ are two consecutive vertices of L_1 and $L_{2.j}$ $L_{2.j+1}$ are two consecutive vertices of L_2. Similarly for the divide distance mode, the calibration pair following ($L_{1.i'}$, $L_{2.j'}$) are ($L_{1.i'+1}$, $L_{2.j'+1}$). $L_{1.i'}$ $L_{1.i'+1}$ are two consecutive points derived from the divide distance of L_1 and, $L_{2.j'}$ $L_{2.j'+1}$ are two consecutive points derived from the divide distance of L_2.

3 Data Matching Between Points of Homologous Geometries

To obtain a high-quality merging process, several pairs of matching points are required using **data matching** processes. These allow one to identify groups of homologous objects that represent the same part of the real world from two sets of geographic data. To obtain an accurate data matching between homologous objects, three different types of data matching must be combined (Devogele *et al.* 1996):

- **Semantic** matching puts objects in correspondence according to their semantic attributes, which are discriminated (Cohen 2000) (Spéry *et al.* 2001). For example, the values of the attributes “identifier of a road” can be used to match roads from two sets of data.
- **Topologic** matching uses composition or topologic relationships between the different objects to match a given object. If two relationships correspond, then this correspondence can be used to find homologous objects linked by this relationship.
- **Geometric** matching employs location of data. Commonly, some measures of distance between objects are computed. Other geometric characteristics, such as the direction (Gabay and Doytsher 1994) have been proposed to match these data. This kind of matching process is the most popular.

In this article, the data matching objective is to identify homologous points (vertex or intermediate point) for geometry of homologous objects. These homologous objects must be found beforehand. Therefore, we focus only on the geometric data matching between points of homologous geometries (points of lines or points of borderlines of polygons.

3.1 Discrete Fréchet Distance

The Fréchet distance is the better maximal linear distance to match a set of ordered points as linear geometries of homologous objects.. For details see for example (Alt and Godau 1995) for a discussion.

The Fréchet distance is the maximal distance between two oriented lines. Each oriented line is equivalent to a continuous function f: [a, a']→V (g: [b, b']→V) where a, a', (b, b') ∈ ℜ, a < a' (b<b') and (V, d) is a metric space. Then d_F denotes their Fréchet distance defined as:

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

An illustration of the Fréchet distance follows: a man is walking with a dog on a leash. He is walking on the one curve, the dog on the other one. Both may vary their speed, but backtracking is not allowed. Then the Fréchet distance of the curves is the minimal length of a leash that is necessary. The Fréchet method has the advantage of computing distances only on a limited number of homologous points.

Eiter and Mannila (Eiter and Mannila 1994) give a good approximation: the discrete Fréchet distance (d_{dF}) that computes in time O(n m). L_1 and L_2 are interpreted as two oriented sets of vertices: <$L_{1.1}$…$L_{1.n}$> and <$L_{2.1}$...$L_{2.m}$>. While d_{dF} is the minimal length of leash i.e. away from the pair of beginning vertices ($L_{1.1}$,$L_{2.1}$) to the pair of ending vertices ($L_{1.n}$,$L_{2.m}$).. This gives an ordered set of ($L_{1.i}$,$L_{2.j}$) such as the following pair of ($L_{1.i}$,$L_{2.j}$) which is one of these three pairs: - ($L_{1.i+1}$,$L_{2.j+1}$) man and dog are walking, - ($L_{1.i+1}$,$L_{2.j}$) only the man is.

To identify a set of homologous points, we focus on vertices for three reasons. The number of homologous point must be limited, the vertices are more accurate than intermediate points (Veregin 1999) and semantically more important. So, we can interpret L_1 and L_2 as two order sets of vertices: <$L_{1.1}$…$L_{1.n}$> and <$L_{2.1}$...$L_{2.m}$>. The discrete Fréchet between L_1 and L_2 is computed recursively as the followings:

$$d_{Fd}(L_1,L_2) = \max\left(\begin{array}{l} d_E\left(L_{1.n}, L_{2.m}\right) \\ \min\left(\begin{array}{l} d_{Fd}\left(< L_{1.1}...L_{1.n-1} >, < L_{2.1}...L_{2.m} >\right) \forall n \neq 1 \\ d_{Fd}\left(< L_{1.1}...L_{1.n} >, < L_{2.1}...L_{2.m-1} >\right) \forall m \neq 1 \\ d_{Fd}\left(< L_{1.1}...L_{1.n-1} >, < L_{2.1}...L_{2.m-1} >\right) \forall n \neq 1, m \neq 1 \end{array}\right) \end{array}\right)$$

<$L_{1.1}$…$L_{1.n-1}$> and <$L_{2.1}$...$L_{2.m-1}$> represent lines. Hence, it is possible to recursively apply this d_{dF} process with parameters: <$L_{1.1}$…$L_{1.n-1}$> , <$L_{2.1}$...$L_{2.m-1}$>… This process is finished when both lines are reduced to both points (<$L_{1.1}$>, <$L_{2.2}$>) and d_{dF} (<$L_{1.1}$>, <$L_{2.2}$>) = d_E($L_{1.1}$, $L_{2.2}$).

For the example in Fig. 3, the matrix of d_E between points $L_{1.i}$ and $L_{2.j}$ is given in table 1 to compute and visualise d_{dF}(L_1, L_2). Note that d_{dF} is equal to 1.90.

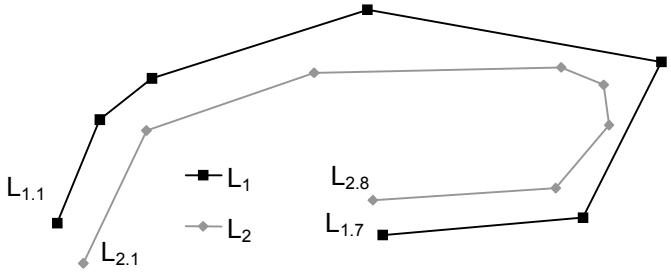

Fig. 3. Example of homologous lines

Table 1. Matrix of d_E between ($L_{1.i}$, $L_{2.j}$) distance and ways

L1	L2	L2.j.x	0.5	1.7	4.9	9.6	10.4	10.5	9.5	6
		L2.j.y	0	2.3	3.3	3.4	3.1	2.4	1.3	1.1
L1.i.x	L1.i.y	i. j	1	2	3	4	5	6	7	8
0	0.7	1	**0.86**	2.33	5.55	9.97	10.67	10.64	9.52	6.01
0.8	2.5	2	2.52	**0.92**	4.18	8.85	9.62	9.70	8.78	5.39
1.8	3.2	3	3.45	**0.91**	3.10	7.80	8.60	8.74	7.93	4.70
5.9	4.4	4	6.97	4.70	**1.49**	3.83	4.68	5.02	4.75	3.30
11.5	3.5	5	11.54	9.87	6.60	**1.90**	**1.17**	**1.49**	2.97	6.00
10	0.8	6	9.53	8.43	5.68	2.63	2.33	1.68	**0.71**	4.01
6.2	0.5	7	5.72	4.85	3.09	4.47	4.94	4.70	3.40	**0.63**

d_{dF} can be used because the length of the longer segment (LengthMaxSeg) (Eiter and Mannila 1994) limits the generated approximation:

$$d_F(L_1, L_2) \leq d_{dF}(L_1, L_2) \leq d_F(L_1, L_2) + \text{LengthMaxSeg}$$

A sampling can be applied to both lines to limit this approximation to ε,. New intermediary vertices can be added such as the length of each segment is inferior to ε (cf. Fig. 5 for an example of sampling). In our case, samplings are needed when the length of segments is important and when the resolutions of the data sets are different.

3.2 Data Matching of Line's Points

d_{dF} are computed to measure the maximal distance between two lines. Hence, d_{dF} can be combined with other processes to match homologous lines. We propose to re-use d_{dF} to define a data matching process between vertices from homologous lines. Indeed, One of d_{dF} ways can be chosen to match the vertices. After using a maximal criteria (d_{dF}), an average criteria is employed. The chosen way, so-called the minimal way (W_m), is defined by the case in which the average distances between its pair ($L_{1.i}$, $L_{2.j}$) is minimal. In other words, between all ways, the one, which has the less taut leash, is chosen.

For the homologous lines of Fig. 3, three ways (the grey cells of table 1 give the pairs of these ways) are possible:

- **W_1**: ($L_{1.1}$,$L_{2.1}$) ($L_{1.2}$,$L_{2.2}$) ($L_{1.3}$,$L_{2.2}$) ($L_{1.4}$,$L_{2.3}$) ($L_{1.5}$,$L_{2.4}$) ($L_{1.5}$,$L_{2.5}$) ($L_{1.5}$,$L_{2.6}$) ($L_{1.6}$,$L_{2.7}$) ($L_{1.7}$,$L_{2.8}$) average of d_E between ($L_{1.i}$,$L_{2.j}$) = 1.12
- **W_2**: ($L_{1.1}$,$L_{2.1}$) ($L_{1.2}$,$L_{2.2}$) ($L_{1.3}$,$L_{2.2}$) ($L_{1.4}$,$L_{2.3}$) ($L_{1.5}$,$L_{2.4}$) ($L_{1.5}$,$L_{2.5}$) ($L_{1.6}$,$L_{2.6}$) ($L_{1.6}$,$L_{2.7}$) ($L_{1.7}$,$L_{2.8}$) average of d_E between ($L_{1.i}$,$L_{2.j}$) = 1.14
- **W_3**: ($L_{1.1}$,$L_{2.1}$) ($L_{1.2}$,$L_{2.2}$) ($L_{1.3}$,$L_{2.2}$) ($L_{1.4}$,$L_{2.3}$) ($L_{1.5}$,$L_{2.4}$) ($L_{1.5}$,$L_{2.5}$) ($L_{1.5}$,$L_{2.6}$) ($L_{1.6}$,$L_{2.6}$) ($L_{1.6}$,$L_{2.7}$) ($L_{1.7}$,$L_{2.8}$) average of d_E between ($L_{1.i}$,$L_{2.j}$) = 1.18

Intuitively, the man and the dog can walk only on this grey cell. For example, if the man is on $L_{1.5}$ and the dog is on $L_{2.5}$, two displacements are possible: - the dog is walking to $L_{2.6}$ and the man is standing in $L_{1.5}$ - the man is walking to $L_{1.6}$ and at the same time the dog is walking to $L_{2.6}$. Moreover, the average distances between pairs of W_m can be computed, 1.12 is inferior as 1.14 and 1.18. So W_1 is the W_m (in bold typeface in table 1). Fig. 4 shows pairs of ($L_{1.i}$,$L_{2.j}$) associated with this minimal way.

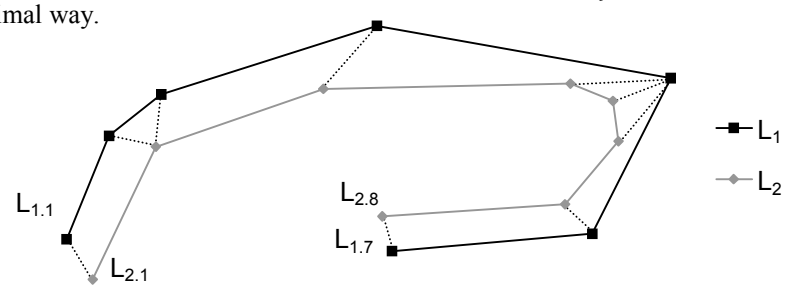

Fig. 4. Pairs of ($L_{1.i}$,$L_{2.j}$) of the minimal way represented by dot lines

We can remark that:

- This matching can be a:
  - one-to-one (between $L_{1.1}$ and $L_{2.1}$ for example)
  - many-to-one (between $L_{1.5}$ and $L_{2.4}$, $L_{2.5}$, $L_{2.6}$ for example)
  - one-to-many (between $L_{1.2}$, $L_{1.3}$ and $L_{2.2}$ for example).

    Generally, a matching of vertices L_1 and L_2 would imply a one-to-one mapping. In our case, many-to-one, or one-to-many matching are not errors, it is only a "turn" with more detail in one data set.
- The average distance method forgives the many-to-many matching.
- Pairs take into account the order of vertex of lines. Graphically, Pairs are a collection of non-crossing dotted lines

This shows that the discrete Fréchet distance can be used to match the points of homologous lines.

3.3 Partial Data Matching of Line's Points

Some pairs of lines, such as the pair shown in Fig.5 can only be partially matched. More precisely, only parts of the lines are matched. For example, still using the lines of Fig. 5, we can visually show that L_1 can be matched to the

partial line <$L_{2.5}$…$L_{2.14}$>. To identify this kind of data matching, d_{dF} cannot be employed. Some other parts of line (<$L_{2.1}$…$L_{2.5}$> and <$L_{2.14}$…$L_{2.17}$>) cannot be used to compute d_{dF}. Therefore, a new measure, so-called the partial discrete Fréchet distance (d_{pdF}), is introduced:

- To detect the partial homologous line < $L_{2.begin}$…$L_{2.end}$ >.
- To compute d_{pdF}. d_{pdF} is equal to d_{dF}(L1, < $L_{2.begin}$…$L_{2.end}$ >) $L_{2.begin}$ and $L_{2.end}$ are chosen such as begin < end and d_{dF}(L1, < $L_{2.begin}$…$L_{2.end}$ >) are smaller. Phase 1 and 2 are simultaneous.
- To chose the minimal way W_m. This phase is similar to the one of data matching between homologous lines.

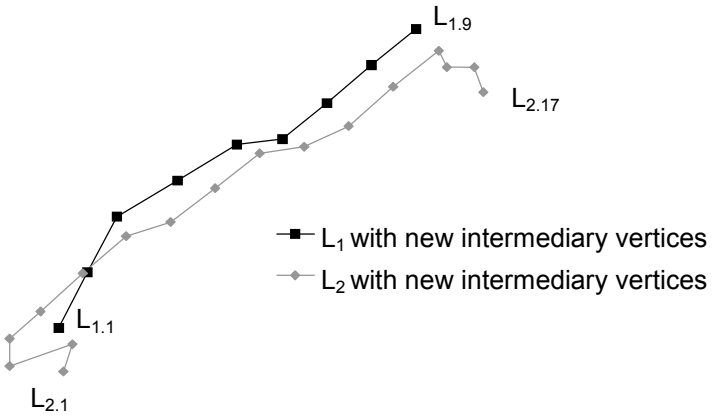

Fig. 5. Example of partial homologous lines with new intermediary vertices

A non-optimal algorithm to compute this measure is:

B = {$L_{2.1}$, $L_{2.2}$, …, $L_{2.m-1}$}; E = {$L_{2.m}$, $L_{2.m-1}$, …, $L_{2.2}$};
d_{Fdp} = + ∞ ;
For $L_{2.j}$ in B
  If d_E($L_{1.1}$, $L_{2.j}$) < d_{Fdp} then
    For $L_{2.jj}$ in E
      If j≤ jj and d_E($L_{1.m}$, $L_{2.jj}$) < d_{Fdp} then
        If d_{Fd}(<$L_{1.1}$…$L_{1.n}$>,<$L_{2.j}$… $L_{2.jj}$>) < d_{Fdp} then
          {d_{Fdp} = d_{Fd}(<$L_{1.1}$…$L_{1.n}$>,<$L_{2.j}$… $L_{2.jj}$>);
          $L_{2.begin}$ = $L_{2.j}$;
          $L_{2.end}$ = $L_{2.jj}$;}

For the lines of Fig. 5, the matrix of d_E between points of $L_{1.i}$ and $L_{2.j}$ is given in table 2 to illustrate the result. Note that d_{pdF} is equal to 1.22. Only one way (grey cells in the matrix) is possible for this example. So the W_m is ($L_{1.1}$,$L_{2.5}$) ($L_{1.2}$,$L_{2.6}$) ($L_{1.3}$,$L_{2.7}$) ($L_{1.3}$,$L_{2.8}$) ($L_{1.4}$,$L_{2.9}$) ($L_{1.5}$,$L_{2.10}$) ($L_{1.6}$,$L_{2.11}$) ($L_{1.7}$,$L_{2.12}$) ($L_{1.8}$,$L_{2.13}$) ($L_{1.9}$,$L_{2.14}$).

Table 2. Matrix of d_E between the partial homologous lines of Fig. 5

	L_2	**L2.j.x**	4.6	4.8	3.4	3.4	4.1	5.05	6	7	8	9	10	11	12	13	13.2	13.8	14
L_1		**L2.j.y**	1.6	2.6	1.8	2.8	3.8	5.15	6.5	7	8.25	9.5	9.75	10.5	11.9	13.2	12.6	12.6	11.7
L1.i.x	L1.i.y	**i.j**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
4.5	3.2	**1**	1.60	0.67	1.78	1.17	**0.72**	2.03	3.62	4.55	6.14	7.74	8.55	9.77	11.4	13.1	12.8	13.2	12.7
5.15	5.2	**2**	3.64	2.62	3.82	2.97	1.75	**0.11**	1.55	2.58	4.17	5.77	6.65	7.89	9.58	11.2	10.9	11.3	10.9
5.8	7.2	**3**	5.73	4.71	5.91	5.01	3.80	2.18	**0.73**	**1.22**	2.44	3.94	4.91	6.16	7.78	9.37	9.16	9.65	9.35
7.15	8.5	**4**	7.36	6.35	7.68	6.82	5.60	3.95	2.31	1.51	**0.89**	2.10	3.11	4.34	5.92	7.50	7.31	7.81	7.56
8.5	9.8	**5**	9.08	8.10	9.49	8.66	7.44	5.79	4.14	3.18	1.63	**0.58**	1.50	2.60	4.08	5.64	5.47	5.99	5.82
9.5	10	**6**	9.72	8.77	10.2	9.44	8.22	6.58	4.95	3.91	2.30	0.71	**0.56**	1.58	3.14	4.74	4.52	5.02	4.81
10.5	11.3	**7**	11.3	10.4	11.8	11.0	9.86	8.22	6.58	5.54	3.94	2.34	1.63	**0.94**	1.62	3.14	3.00	3.55	3.52
11.5	12.7	**8**	13.0	12.1	13.5	12.7	11.5	9.93	8.29	7.26	5.66	4.06	3.31	2.26	**0.94**	1.58	1.70	2.30	2.69
12.5	14	**9**	14.7	13.7	15.2	14.4	13.2	11.5	9.92	8.90	7.30	5.70	4.93	3.81	2.16	**0.94**	1.57	1.91	2.75

Partial data matching is required to match one line to a part of another line. To match a part of a given line to another line, detection of homologous parts of lines is a more complex process. It is always possible to reduce the d_{dF} between parts, by part reduction. This more complex data matching process is not treated in this paper.

3.4 Data Matching of Polygon's Points

This process can also be applied to match vertices of homologous polygon borderlines. However, for oriented lines, the beginning pair of points and the end pair of points are known. Unfortunately, for polygon borderlines, these pairs are not predetermined. Thus, the process must define a function T to translate polygons borderlines P_1 and P_2 into lines L_1 and L_2 such as the d_{Fd} between L_1 and L_2 is minimal. Subscripts of L_1 and P_1 are identical. On the other hand, the L_2 subscripts correspond to P_2 subscripts only by a circular translation (if $L_{2.j'} = P_{2.m}$ then $L_{2.j'+1} = P_{2.1}$ else $L_{2.j'+1} = P_{2.j+1}$).

A new method is defined as follows:

- To find j' such as $<P_{2.j'}, P_{2.j'+1}\ldots P_{2.m}, P_{2.1}\ldots P_{2.j'-1}>$ is the ordered set of vertex of L_2.
- To compute $d_{dF}(L_1, L_2)$, j' is chosen such as $d_{dF}(L_1, L_2)$ is minimal. So, phase 1 and 2 are simultaneous.
- To chose the minimal way (W_m). This phase is similar to the data matching between homologous lines.
- To translate this W_m from L_1, L_2 into P_1,P_2 using the inverse circular translation

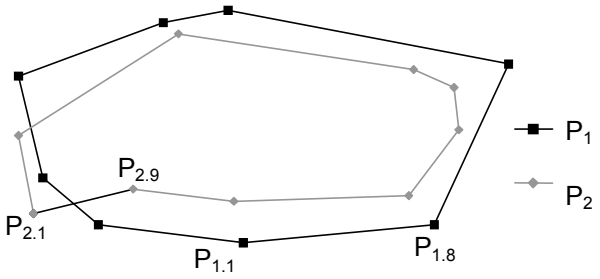

Fig. 6. Example of homologous polygons

Table 3. Matrix of d_E between homologous polygons of Fig. 6

L1	L2	L2.j.x	2	1.7	4.9	9.6	10.4	10.5	9.5	6	4
		L2.j.y	1	2.3	4	3.4	3.1	2.4	1.3	1.2	1.4
L1.i.x	L1.i.y		1	2	3	4	5	6	7	8	9
6.2	0.5	1	4.23	4.85	3.73	4.47	4.94	4.70	3.40	**0.73**	2.38
3.3	0.8	2	1.32	2.19	3.58	6.82	7.46	7.38	6.22	2.73	**0.92**
2.2	1.6	3	**0.63**	0.86	3.61	7.62	8.34	8.34	7.31	3.82	1.81
1.7	3.3	4	2.32	**1.00**	3.28	7.90	8.70	8.85	8.05	4.79	2.98
4.6	4.2	5	4.12	3.47	**0.36**	5.06	5.90	6.17	5.69	3.31	2.86
5.9	4.4	6	5.17	4.70	**1.08**	3.83	4.68	5.02	4.75	3.20	3.55
11.5	3.5	7	9.82	9.87	6.62	**1.90**	**1.17**	**1.49**	2.97	5.96	7.79
10	0.8	8	8.00	8.43	6.02	2.63	2.33	1.68	**0.71**	4.02	6.03

The matrix of d_E between points $P_{1.i}$ and $P_{2.j}$ in Fig. 6 is given in table 3 to visualise the d_{dF} and ways. In this example, j' is equal to 8. So $d_{dF}(P_1, P_2)$ is equal to 1.90 and four ways are possible. Pairs of points are the grey cells of table 1. The value W_m is given in bold typeface.

4 Weighting Function of the Merging Process

The new merging process proposed in this paper, requires the minimal way (W_m) and the weighting function (f_w). The k^{th} vertex of the resulting line (L_M) is defined thanks to the k^{th} pair ($L_{1.i.}$, $L_{2.j}$) of the W_m and f_w, it is defined as follows:

$$L_{F.k} = f_w\left(L_{1.i}, L_{2.j}\right) = \frac{\alpha_{1.k} \times L_{1.i} + \alpha_{2.k} \times L_{2.j}}{\alpha_{1.k} + \alpha_{2.k}}$$

For each pair, the weight ($\alpha_{1.k}$, $\alpha_{2.k}$) is proportional to:

- The accuracy of the input layers,

- The kind of points considered. The weight of a vertex is more important than the weight of a new intermediary vertex. According to (Vergin 1999), vertices are more accurate than the other points of the line.
- The cardinality of the pairs of W_m where this point is included. The weight of a point included in a single pair is more important than the weight of a point included in several pairs. Indeed, a point included in several pairs, represents a curve more detailed in the other layer. The resulting curve must be closer to the more detailed curve.

In the case of the two lines of Fig. 3, the merging line can be computed thanks to their W_m and f_w (cf. table 4). For the example, we suppose L_1 less accurate than L_2. A default weight of 0.8 is associated with the points of L_1 and a default weight of 1 is associated with the points of L_2. For each point, the default weights are divided by the number of pairs in the line.. Weights: $\alpha_{1.5}$, $\alpha_{1.6}$, $\alpha_{1.7}$ associated with $L_{1.5}$ are divided by 3 and weights $\alpha_{2.2}$, $\alpha_{2.3}$ are divided by 2. The merging line is given in Fig. 7.

This example visually shows that this merging process gives a suitable result. In order to define the merging line, this process uses the accuracy of lines and the "local" accuracy. This last one is taken into account by the cardinality of pairs. In the case of Fig. 7, the merging line is:

- Globally closer to L_2 (the more accurate line),
- Locally closer to more accurate than part of line ($L_{2.2}$, $L_{2.3}$ and $L_{1.4}$, $L_{1.5}$, $L_{1.6}$).

Table 4. Merging process of lines of Fig. 3, thanks to the pairs of W_m and f_w

Wm						weight		Merging line	
$L_{1.i}$	$L_{2.j}$	$L_{1.i.x}$	$L_{1.i.y}$	$L_{2.j.x}$	$L_{2.j.y}$	$\alpha_{1.k}$	$\alpha_{2.k}$	$L_{mk.x}$	$L_{mk.y}$
$L_{1.1}$	$L_{2.1}$	0	0,7	0,5	0	0,80	1,00	0,28	0,31
$L_{1.2}$	$L_{2.2}$	0,8	2,5	1,7	2,3	0,80	0,50	1,15	2,42
$L_{1.3}$	$L_{2.2}$	1,8	3,2	1,7	2,3	0,80	0,50	1,76	2,85
$L_{1.4}$	$L_{2.3}$	5,9	4,4	4,9	3,3	0,80	1,00	5,34	3,79
$L_{1.5}$	$L_{2.4}$	11,5	3,5	9,6	3,4	0,27	1,00	10,00	3,42
$L_{1.5}$	$L_{2.5}$	11,5	3,5	10,4	3,1	0,27	1,00	10,63	3,18
$L_{1.5}$	$L_{2.6}$	11,5	3,5	10,5	2,4	0,27	1,00	10,71	2,63
$L_{1.6}$	$L_{2.7}$	10	0,8	9,5	1,3	0,80	1,00	9,72	1,08
$L_{1.7}$	$L_{2.8}$	6,2	0,5	6	1,1	0,80	1,00	6,09	0,83

For this example, all points are vertices, thus, new intermediary vertices are not considered. Empirically, to consider the lower accuracy of the additional point, the weight of these points should be multiplied by 0.8.

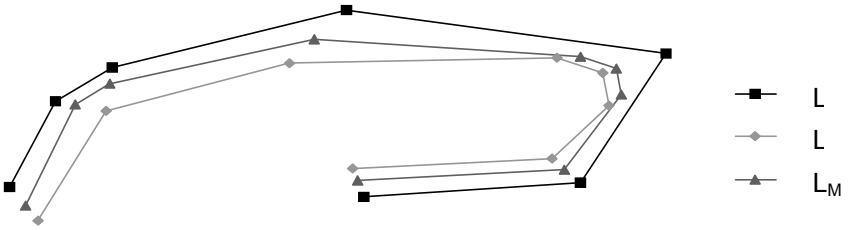

Fig. 7. Merging line obtained by the merging process

To merge partial homologous lines or polygons, the k^{th} vertex of L_M or P_M is also defined thanks to the k^{th} pair and a similar weighting function. Weights are proportional to the three same arguments (accuracy of the input layers, kind of points and cardinality of pairs). Fig. 8 shows the resulting line L_M (see (a)) and the polygon P_M (see (b)) computed for our examples.

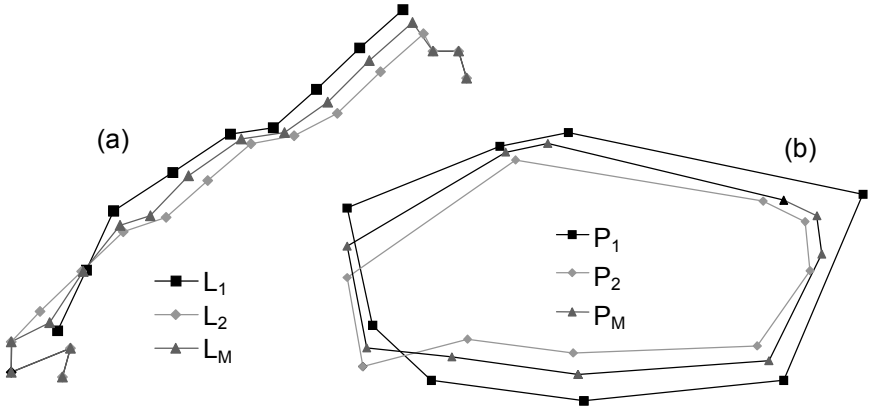

Fig. 8. Merging line obtained by the partial merging process and merging polygon obtained by the merging process

5 Joining Process

When the merging process transforms a point ($L_{i,j}$) into several points ($L_{M,j'}$, $L_{M,j''}$…) there is a join problem at the line (L_k) which is connected to $L_{i,j}$. To preserve the topology, L_k must be joined with one point of L_M.

Two situations are distinguished:

- $L_{i,j}$ is transformed into an odd number of points. The average point is used to join L_k (see Fig. 9a)
- $L_{i,j}$ is transformed into an even number of points. A new point is defined at the halfway of the middle edge (see Fig. 9b).

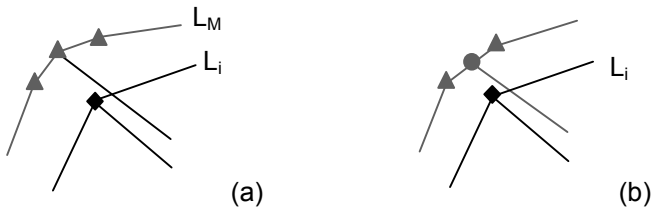

Fig. 9. Examples of junction between the merging line and another line

6 Conclusion

This paper introduces a new merging process that can be an alternative to fuzzy tolerance methods for a better data integration of homologous objects. It makes the distinction between different and homologous objects, thanks to data matching and measures derived from the Fréchet distance. Thus, homologous objects are merged and different objects are overlaid.

Moreover, it uses homologous points and not closest points to compute merging lines. Another advantage of this merging process relies on the fact that it can be automatic and generic. In others words, it can be applied to homologous lines, homologous polygons and partial homologous lines. Additionally, this process takes into account connections of merging lines to other lines.

This approach must be combined, however, with other data integration processes like rubber sheeting (to smooth displacements induced by the merging and joining processes). Any data integration based on this novel merging (e.g. overlaying, merging, joining and rubber sheeting) must be organised in terms of data and process schedule. The proposed merging technique is defined to integrate data from different layers in the same geographic area, it can also be extended to integrate data from adjacent coverages.

References

Alt H, and Godau M (1995) Computing the Fréchet distance between two polygonal curves. International Journal of Computational Geometry & Applications 5(1-2):75-91

Chrisman N (2001) Exploring Geographic Information Systems, 2nd Edition. John Wiley & Sons, Inc.

Cohen W (2000) Data Integration using similarity joins and a Word-Based Information representation language. ACM Transactions on Information Systems 18(3):288-321

Devogele T, Trevisan J, Raynal L (1996) Building a Multi-Scale Database with Scale-Transition Relationships. In: Kraak et Molenaar (eds) Spatial Data Handling. pp 337-351

Devogele T, Parent C, Spaccapietra S (1998) On spatial database integration. International Journal of Geographical Information Science 12(4):335-352

Dougenik JA (1980) WHIRLPOOL: A geometric processor for polygon coverage data. In: 4th Auto-Carto. pp 304-311

Harvey F, Vauglin F (1996) Geometric Match-processing: Applying Multiple Tolerances. In: Kraak and Molenaar (eds) Spatial Data Handling. pp 155-171

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

Parent C, Spaccapietra S, Devogele T (1996) Conflicts in Spatial Database integration. In: Parallel and Distributed Computing Systems. pp 772-778

Pullar D (1993) Consequences of using a tolerance paradigm in spatial overlay. In: 11th Auto Carto. pp 288-296

Spéry L, Claramunt C, Libourel T (2001) A Spatio-Temporal Model for the Manipulation of Lineage Metadata. GeoInformatica 5(1):51-70

TCI software (1999) Adjust : True Rubber Sheeting inside AutoCAD [online]. Baker City, Oregon. Available from: http://www.tcicorp.com/tci_adjust.html

USGS (1998) About the USGS Geospatial Data Clearinghouse [online]. Available from: http://nsdi.usgs.gov/

Veregin H (1999) Data quality parameters. In : Longley, Goodchild, Maguire, Rhind (eds) Geographical Information Systems. John Willey & Sons, pp 177-189