International Journal of Geographical Information Science
Vol. 00, No. 00, Month 200x, 1–26

RESEARCH ARTICLE

An Inconsistency Measure of Spatial Data Sets with respect to Topological Constraints

Nieves R. Brisaboa[a], Miguel R. Luaces[a], M. Andrea Rodríguez[b], Diego Seco[ab*]
[a]*Database Laboratory, University of A Coruña, Spain*
[b]*Department of Computer Science, University of Concepción, Chile*

(*Received 00 Month 200x; final version received 00 Month 200x*)

An inconsistency measure can be used to compare the quality of different datasets and to quantify the cost of data cleaning. In traditional relational databases, inconsistency is defined in terms of constraints that use comparison operators between attributes. Inconsistency measures for traditional databases cannot be applied to spatial datasets because spatial objects are complex and the constraints are typically defined using spatial relations. This paper proposes an inconsistency measure to evaluate how dirty a spatial dataset is with respect to a set of integrity constraints that define the topological relations that should hold between objects in the dataset. The paper starts by reviewing different approaches to quantify the degree of inconsistency and showing that they are not suitable for the problem. Then, the inconsistency measure of a dataset is defined in terms of the degree in which each spatial object in the dataset violates topological constraints and where possible representations of spatial objects are points, curves, and surfaces. Finally, an experimental evaluation demonstrates the applicability of the proposed inconsistency measure and compares it with previously existing approaches.

Keywords: Inconsistency measure; topological similarity measure; spatial inconsistency

1. Introduction

Inconsistency is a kind of data imperfection. It refers to a contradiction with a model of the reality that is typically expressed by a set of *integrity constraints*. Although con-

*Corresponding author. Email: dseco@udec.cl

ISSN: 1365-8816 print/ISSN 1362-3087 online

DOI: 10.1080/13658816.2013.811243
http://www.informaworld.com

sistency is a desirable property of a dataset, it is common to find inconsistency due to errors introduced during data manipulation and integration processes.

Traditionally, inconsistency in traditional databases has been a binary property, i.e., the database is either consistent or not. Recent work defines *inconsistency measures*, also known as *dirtiness measures*, that count the number of elements that violate integrity constraints or define variance measures in symbolic or numerical data (Martinez *et al.* 2007). Most common measures check if each element in the database is consistent or not and, therefore, consistency is a binary property of the elements in the database instead of being a property of the database itself. The domain of spatial databases, however, rises new issues because spatial objects are complex data and, therefore, they are no longer simply *consistent* or *inconsistent* with respect to an integrity constraint. In addition, constraints are typically defined in terms of spatial relations between objects, in particular *topological predicates*, which implies that more complex functions need to be defined to evaluate the difference between the actual topological relation between objects and their expected topological relation as expressed by the topological constraints.

The following example serves as a motivation of inconsistency measures in the spatial domain.

Example 1.1 Consider a database instance[1] that stores administrative boundaries such that any two boundaries must either be disjoint or they must touch each other. Geometries of a database instance with five boundaries are shown in Figure 1. This database is inconsistent since g_3 intersects g_1 and g_2, and g_4 intersects g_2 and g_5. A binary qualification of inconsistency would consider that all geometries in the database violate the integrity constraint (i.e., 100% of inconsistency). On the other hand, an inconsistency measure would compute a degree of inconsistency for the database. For example, it could compute the sum of the relative area of each geometry that overlaps any other region in the set, which sums up to much less than 100% of the total area of the geometries in the database. Even more, an inconsistency measure associated with spatial objects could indicate that geometry g_4 has a larger area in conflict than any other geometry in the database, which may guide the cleaning process of the database.

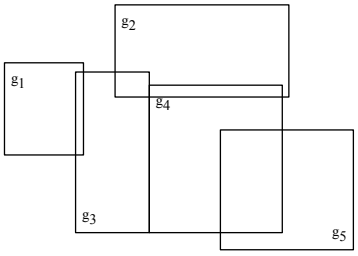


Figure 1. An example of an inconsistent dataset: boundaries must either be disjoint or touch.

□

There are many useful applications of inconsistency measures. An inconsistency measure allows us to quantify the data quality of a dataset, and consequently, to compare the quality of different data sources in an integration process. An inconsistency measure is also useful when analyzing manipulation processes that lead to potential inconsistency

[1]A database instance is a dataset that complies a data model

situations, such as generalization. Another example is the case of data cleaning. In this kind of process, the geometries of objects are modified in order to restore consistency. One approach could be selecting and correcting an object that violates an integrity constraint at random, which reduces the inconsistency, but it may not be efficient in terms of reducing the number of objects to be modified. Even more, this approach does not guide the process to make large reductions of inconsistency with the minimum number of changes. Inconsistency measures can be used to define strategies to guide the cleaning process by determining the most appropriate ordering for the correction of geometries. As a summary, in these types of applications, the fundamental idea is to compare what the data is expected to be from what the data is indeed, and inconsistency measures provide an objective method to evaluate this comparison.

In a previous work (Rodríguez *et al.* 2010), we defined a set of measures that evaluate the degree of conflict of pairs of objects with respect to *topological constraints*, also known as topological dependency constraints (Bravo and Rodríguez 2012) or topological integrity constraints (Hadzilacos and Tryfona 1992). These measures associate a violation degree with pairs of objects, being equal to 0 when no violation occurs. We evaluated these measures by comparing them to the semantic distance in a conceptual neighboring graph of topological relations (Egenhofer and Al-Taha 1992, Egenhofer and Mark 1995), and to the distance between boundary points of geometries in a consistent dataset that was made artificially inconsistent. Both tests showed that our measures, in addition to being correlated to other measures, provide metric details about the degree in which two geometries are in conflict. Then, the basic criteria used to define these measures were validated with a human-subject testing by Brisaboa *et al.* (2011) , and we discovered that the external distance, the length of the crossing segments, and the size of the overlapping area were the main factors used by the subjects to evaluate the violation degree of a topological relation. We also discovered that the internal distance and the relative size of geometries in conflict with respect to other geometries has less impact on the evaluation of the violation degree.

In this paper we summarize our previous results and we propose and analyze a new set of measures of the violation degree of spatial objects that overcome problems detected by our previous proposal. These new measures take a different approach. Instead of considering pairs of objects to compute the degree of violation of a topological relation, they now quantify the degree of violation of an object with respect to all other objects in the dataset. Thus, we associate the degree of violation with each individual object instead of to every pair of objects. This approach has at least two advantages. On the one hand, we can define the degree of violation of an object in a set, and in doing so, we can consider that all violations of the object are not necessarily independent of each other. On the other hand, we can use these measures to define cleaning strategies that are object-oriented.

The organization of the paper is as follows. Section 2 presents previous approaches to define inconsistency measures and compare topological relations. Section 3 provides basic concepts and summarizes the criteria to define measures that were validated by Brisaboa *et al.* (2011). Section 4 presents the proposed inconsistency measure of spatial datasets based on a new set of measures that quantify the violation degree of objects. An experimental evaluation is given in Section 5, and final conclusions and future research directions are given in Section 6.

2. Related work

The notion of inconsistency relates to the notion of data quality, which is a broader concept that is also associated with other types of data imperfection, such as uncertainty, vagueness, and incompleteness of datasets (Bonnissone and Tong 1985, Bosc and Prade 1997, Parson 1996). Spatial inconsistency refers to a contradiction between stored data and *structural* or *semantic* constraints. For example, specifying that a surface must be bounded by closed and non self-intersecting polylines is a structural constraint, whereas specifying that two land parcels must be internally disconnected is a semantic constraint. Some important properties of spatial data that show the propensity of spatial datasets to be inconsistent were described by Plümer and Gröger (1997), and Borges *et al.* (2002):

(i) Many spatial data are inherently vague, which may lead to conflicting data.
(ii) Topological and other spatial relations are very important and are usually implicitly represented. Spatial relations are typically derived through data manipulation.
(iii) Spatial datasets may need to treat different levels of detail in the spatial representation. These representations may be handled as duplicate information or may be generated dynamically through a generalization process.

Related work addressing (in)consistency measures in databases is found in the relational context. A strategy to measure consistency with respect to referential integrity constraints in distributed databases is described by Ordonez *et al.* (2007). It defines both local and global measures of consistency and completeness of data, from consistency of tables to consistency of the database. The work in Martinez *et al.* (2007) presents a set of dirtiness measures for inconsistency numerical data with respect to functional dependency constraints: number of tuples that violate a constraint, number of violations (several tuples may be related to one single violation), rate between the number of violations and the tuples in the database, and variance function between numerical values. They also define a set of axioms upon which different measures were compared. In a similar way, Motro and Rakov (1996), and Jiang *et al.* (2009) estimate the quality of databases in terms of *soundness* and *completeness*, which are similar to precision and recall in the context of information retrieval (Baeza-Yates and Ribeiro-Neto 1999). These proposals are not easily generalizable to the context of spatial databases due to the complexity of spatial operators and objects.

In the context of spatial databases, a study by Rodríguez *et al.* (2013) introduces the notion of *repair semantics* to define consistent query answers from inconsistent databases. This repair semantics defines a systematic way to repair inconsistency with respect to topological constraints expressed as denial constraints. The work also defines a distance measure to compare different alternative repairs and to obtain a repair that minimally differs from the original database. In principle, one could use the concept of repair semantics and apply it to an inconsistent database. Then, the inverse of the distance between the repair and the original inconsistent database defines a consistency measure because it represents the effort needed to restore consistency. The main problem of this approach is that there is potentially an exponential number of repairs and the problem of deciding whether a modified database is a repair that minimally differs from an original inconsistent database is computationally intractable (Rodríguez *et al.* 2013).

Related work also addresses similarity measures of topological relations. Similarity measures are useful to compare the topological relation between objects with respect to a topological constraint. Using a qualitative approach to define topological relations (Egenhofer and Franzosa 1991, Randell *et al.* 1992b), a similarity measure compares topolog-

ical relations using the semantic distance between relations as defined by a conceptual neighborhood graph (Papadias *et al.* 1998) (Figure 2). This type of measure suffers the disadvantage that it does not distinguish pairs of objects that, holding the same topological relation, correspond to different spatial configurations. For example, it does not distinguish between a pair of disjoint objects very close to one another and a pair of disjoint objects very far apart.

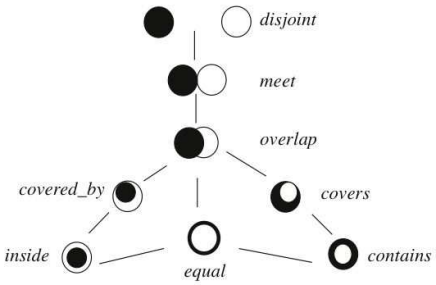


Figure 2. Conceptual neighborhood of topological relations (Egenhofer and Al-Taha 1992).

Another similarity measure of topological relations defines similarity as the inverse of the difference between the distance and angle of the centroids of objects (Berreti *et al.* 2000). A different study defines ten measures that characterize topological relations based on metric properties, such as *length*, *area*, and *distance* (Egenhofer and Shariff 1998, Egenhofer and Dube 2009). The combination of these measures gives an indication of the topological relation between objects. Using the Minimum Bounding Rectangle (MBR) of objects, the work in Godoy and Rodríguez (2004) characterizes topological relations as combinations of overlapping areas and distances between objects. The problem of using the previous measures for our purposes is that all of them compare relations between pairs of objects (i.e., geometries). In our case, in contrast, we need to compare the topological relation that objects hold with respect to the topological relation they should hold.

3. Preliminaries

3.1. *Topological constraints*

Data inconsistency is an indication of the agreement of data with respect to a model of reality. This model is usually expressed by a set of integrity constraints. We concentrate our work on constraints that impose topological relations depending on the semantics of spatial objects. Topological relations have spurred much research (Egenhofer and Franzosa 1991, Randell *et al.* 1992a) and, since they are implemented in current Spatial SQL, they are useful not only for querying databases but also for expressing integrity constraints.

This work focuses on the subset of topological relations for 2D objects, which are currently implemented in spatial query languages proposed by ISO (ISO 2004) and the Open Geospatial Consortium (OGC 2011). The analysis for topological relations between 3D spatial objects (Zlatanova *et al.* 2004, Lee and Kwan 2005) has been left as future work. The set of topological relations that we consider here includes a subset of base relations for regions (Egenhofer and Franzosa 1991, Randell *et al.* 1992a), and it also includes derived relations such as Intersects, Within, and Contains, which are defined as a conjunction of base relations.

Table 1. Definition of topological relations by the Open Geospatial Consortium (OGC 2011).

Relation	**Definition**
Disjoint(x, y)	True if $x \cap y = \emptyset$
Touches(x, y)	True if $x \cap y \subseteq (\partial(x) \cup \partial(y))$
Equals(x, y)	True if $x = y$
Within(x, y)	True if $x \subseteq y$
Contains(x, y)	True if $y \subseteq x$
Overlaps(x, y)	True if $x \cap y \neq \emptyset$, $x \cap y \neq x \neq y$, and $dim(x \cap y) = dim(x) = dim(y)$
Crosses(x, y)	True if $x \cap y \neq \emptyset$, $x \cap y \neq x \neq y$, and $dim(x \cap y) < max(dim(x), dim(y))$
Intersects(x, y)	True if $x \cap y \neq \emptyset$

Table 2. Possible relations between geometries of different dimensions (Surface, Curve, Point). For legibility, we skip relations C×P and P×C, which are equivalent to S×P and P×S, respectively.

Relation	S×S	C×C	P×P	S×C	C×S	S×P	P×S
Disjoint	√	√	√	√	√	√	√
Touches	√	√		√	√		
Overlaps	√	√					
Within	√	√			√		√
Contains	√	√		√		√	
Crosses		√		√	√		
Intersects	√	√		√	√	√	√
Equals	√	√	√				

Table 1 provides the definitions of the topological relations used in this work, which were extracted from the Open Geospatial Consortium Simple Feature Specification (OGC 2011). In this table, given a geometry x, $\partial(x)$ indicates its boundary, and $dim(x)$ its dimension, where $dim(x)$ is equal to 0 if x is a point, 1 if it is a curve, and 2 if it is a surface.

Based on the definition of topological relations, the possible relations between geometries of different dimensions in a 2D space are shown in Table 2. Notice that relations between curves and surfaces are not symmetric. For example, a curve can be Within a surface but this is not true for a surface with respect to a curve. This is similar for a point with respect to a curve or surface.

In terms of the specification of spatial integrity constraints that concern topological relations, previous studies define topological constraints (Borges *et al.* 1999) and spatial semantic constraints (Mäs 2007, Hadzilacos and Tryfona 1992). Servigne *et al.* (2000) specify constraints that introduce explicitly four types of cardinalities: forbidden, at least t times, at most t times, and exactly t times. Consequently, one could express, for example, that a sluice joins a water pipe exactly two times. More recently, types of spatial semantic integrity constraints and an analysis of the database consistency problem with these constraints were presented by Bravo and Rodríguez (2012). These types of integrity constraints combine classical functional and referential integrity constraints of relational databases with topological relations and check constraints (e.g., the stored area of a geometry should be equal to the area calculated from the stored geometry).

The work in this paper addresses consistency with respect to topological constraints (TCs) that define the topological relationship T that should hold between any two objects (for example, two land-parcels a, b should touch each other). Hence, this constraint is violated when a relationship different from T holds between such objects (for example, when a overlaps b, or a is disjoint from b). Notice that T may be a basic or a derived topological relation defined by a disjunction of basic topological relations.

More formally, let T be a topological relation or a conjunction of topological relations, $R_1(x_1, \ldots, x_n, g_1)$ and $R_2(y_1, \ldots, y_m, g_2)$ be predicates representing types of spatial ob-

jects (e.g., land parcel or building) in a database, where x_i and y_i are alphanumeric attributes, and g_1 and g_2 are geometric attributes. The topological constraint addressed by this work is of the form:

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_m, g_1, g_2 \; (R_1(x_1, \ldots, x_n, g_1) \wedge R_2(y_1, \ldots, y_m, g_2) \wedge \psi \rightarrow T(g_1, g_2)) \quad (1)$$

where ψ is an optimal conjunctive formula of the form $w_1 \neq z_1 \wedge \cdots \wedge w_l \neq z_l$ with $w_i \in \{x_1, \ldots, x_n\}$ and $z_i \in \{y_1, \ldots, y_m\}$, for all $i \in [1 \ldots l]$. The inclusion of ψ is used to express constraints where R_1 and R_2 represent the same type of spatial object (e.g., land parcel) and we want the variable to instantiate different objects.

Example 3.1 Consider a predicate with scheme $\mathsf{LandParcel}(id, g)$, a constraint could be "land parcels cannot internally intersect", which can be expressed in the form of Equation 1 as:

$$\begin{aligned} \forall id_1, id_2, g_1, g_2 \; (&\mathsf{LandParcel}(id_1, g_1) \wedge \mathsf{LandParcel}(id_2, g_2) \wedge \\ &(id_1 \neq id_2) \rightarrow \mathsf{Touches}(g_1, g_2) \vee \mathsf{Disjoint}(g_1, g_2)). \end{aligned} \quad (2)$$

□

A database instance D violates a constraint of the form (1) when there are tuples $R_1(a_1, \ldots, a_n, u_1) \wedge R_2(y_1, \ldots, y_m, u_2)$ in D such that $R_1(a_1, \ldots, a_n, u_1) \wedge R_2(b_1, \ldots, b_m, u_2) \wedge \psi$ is true but $T(u_1, u_2)$ is false. We will say in this case that u_1 and u_2 are in conflict with respect to topological relation T.

This definition of topological constraints does not allow the expression of all possible constraints. For example, the following constraint in a database model cannot be expressed with our definition of topological constraints: "a building must not be closer than 100 meters to any road". To express this constraint, one would need to use geometric operators such as *distance* or *buffer* to specify the separation that should hold between buildings and roads. However, our simple definition can be used to express many meaningful constraints and captures the essence of what the inconsistency measures proposed in this work quantify. Including additional geometric operators would only require to extend the definitions of the inconsistency measures.

3.2. *Definition factors of inconsistency measures*

Rodríguez *et al.* (2010) presented different measures that compare topological relations between geometries stored in a database instance with respect to expected topological relations. The definition of all measures is based on two aspects: the difference between the actual and the expected relation between objects, and the relevance of the objects in the dataset. The measures are defined only for the combination of two surfaces or two curves.

The measures are used to give a global evaluation of the data quality with respect to topological constraints in the following way. First, we define the concept of *checked topological relationships (CTR)* as the subset of pairs of objects that have to be analyzed to check the topological constraint. Then, we define the *violation spread (VS)* to compute how many pairs of objects violate a topological constraint over the total number of CTR,

and the *global fulfillment (GF)* to consider not only the number of violations but also their importance.

The measures defined in (Rodríguez *et al.* 2010) have two important drawbacks:

(i) They are computed in a pair-wise manner, and therefore if more than two spatial objects are involved in the same conflict, the violation is reported several times.
(ii) We need to predefine or compute the set of *checked topological relationships*, since otherwise, we should check any pair of possible objects with the consequent impact on the efficiency of computation and on the global measure of the dataset.

In this work, we define measures that assign a degree of inconsistency to each object taking into account all the conflicts where it is involved. Hence, we avoid the problem of reporting the same violation several times. Moreover, we detect which objects violate the constraints and we do not have to use the concept of checked topological relationships. Furthermore, we extend the definitions to include relations between a curve and a surface, between a point and a curve, and between a point and a surface.

In (Brisaboa *et al.* 2011), we perform a cognitive validation of the factors that affect the degree of violation of objects with respect to a topological relation. Five different factors with respect to different topological relations were analyzed:

(i) The *external distance* between disjoint geometries, which has an impact on conflicts risen by the separation of geometries when they must intersect.
(ii) The *internal distance* between geometries when one geometry is within the other, which has an impact on conflicts when geometries must be externally connected (i.e., they must touch or they must be disjoint).
(iii) The *overlapping area* of geometries that are internally connected, which has an impact on conflicts when geometries must be externally connected.
(iv) The *crossing length* that represents the length of the minimum segment of a curve that crosses another curve or a surface, which has an impact on conflicts when geometries must be externally connected.
(v) The *size difference* that represents how much a geometry must grow to become equal or contain another geometry.

The results show that *external distance*, *overlapping area*, and *crossing length* are consistently relevant factors to evaluate the violation degree of an object. Regarding *internal distance*, the results do not support nor reject its influence in the perception of the violation degree by human subjects. Finally, the *touching length* was rejected as a useful factor to evaluate the degree of violation of a topological relation. Furthermore, the results conclude that the relative size of the geometries within the dataset is not considered to be important by the subjects, or at least, that the subjects consider more important the magnitude of the conflicts than the relative size of the geometries with respect to other geometries in the dataset.

In this work, we take into account the results of the cognitive validation in the definition of the measures. We no longer use the relative sizes of the geometries and we simplify the measures by using the overlapping area and the external distance as the main factors in the computation of the violation degrees.

4. Inconsistency and violation degree measures

4.1. *General concepts*

In this section, we define an inconsistency measure of spatial datasets in a top-down manner. Let D be a spatial database instance that includes a collection of tuples with a spatial attribute whose values are in the set $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{G}|}\}$ and $\mathcal{TC}$ be the set of topological constraints defined over D. Then, the *inconsistency measure* of a database $InC(D, \mathcal{TC})$ is defined as follows.

$$InC(D, \mathcal{TC}) = \frac{\sum_{i=1}^{|\mathcal{G}|} VD(g_i, D, \mathcal{TC})}{|\mathcal{G}|} \tag{3}$$

Given a spatial object $g \in \mathcal{G}$, the violation degree $VD(g, D, \mathcal{TC})$ quantifies the degree in which the geometry g participates in the violation of the topological constraints in $\mathcal{TC}$ with other geometries in D. Let $VD_{Tc_i}(g, D)$ be the violation degree of an object with respect to a specific topological constraint $Tc_i \in \mathcal{TC}$, the violation degree of a spatial object in the database D with respect to $\mathcal{TC}$ can be defined as follows:

$$VD(g, D, \mathcal{TC}) = min(1.0, \sum_{\forall Tc \in \mathcal{TC}} VD_{Tc}(g, D)) \tag{4}$$

We have defined $VD(g, D, \mathcal{TC})$ in such a way that the violation degree always takes values between zero and one. A violation degree of zero means that the object does not violate any topological constraint, and a value of one means that the whole geometry of the object is considered to participate in the violation of topological constraints. We define next the violation degree of a spatial object with respect to a specific topological constraint. To simplify the notation we will say that a topological relation T is in a topological constraint Tc of the form (1) ($T \in Tc \in \mathcal{TC}$) when Tc imposes a topological relation T.

Let g be a spatial object and $\mathcal{G}' \subseteq \mathcal{G}$ such that g must hold a topological relation T with each spatial object $g_j \in \mathcal{G}'$, but instead it holds a relation $T' \neq T$. The violation degree of g with respect to T is determined by the aggregation of the violation degree of g and each spatial object $g_j \in \mathcal{G}'$ (denoted as $VD_T^{g_j}(g)$). Many times the violation degrees of g with each spatial object in $\mathcal{G}'$ are independent, and thus, the arithmetic sum can be used to compute the aggregation:

$$VD_{Tc}(g, D) = min(1.0, \sum_{j=1}^{|\mathcal{G}'|} VD_{T \in Tc}^{g_j}(g)) \tag{5}$$

However, this is not the case for all kind of topological constraints because the intersections of geometries are not necessarily disjoint and this should be considered to avoid counting more than once the overlapping areas. The following example illustrates this idea.

Example 4.1 Consider the following set of 3 surfaces that should touch each other.

Note that they may represent land-parcels, counties, states, etc. We want to define the violation degree associated with geometry g that overlaps geometries g' and g'', which also overlap each other (see Figure 3). A simple approach to define the violation degree associated with g could add all conflicts independently (as in Equation 5). However, this approach counts the overlapping area among g, g', and g'' more than once. Instead, one could consider the violation degree of g as a factor of the overlapping area of g with the geometric union of all overlapping geometries (i.e., $\mathcal{G}' = \{g' \cup g''\}$). However, this approach is computationally more expensive.

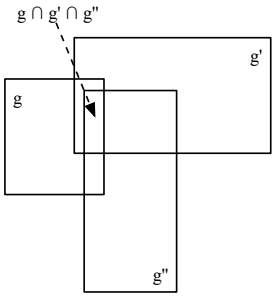


Figure 3. An example of an inconsistent dataset with sharing overlapping area.

□

Therefore, there is a trade-off between efficiency and precision because the use of the geometric union is computationally expensive and it should be replaced by the arithmetic sum when possible. However, a characterization of the cases when geometric union is necessary is out of the scope of this work, and we leave it as an open problem.

Before presenting the definition of the violation degree for each T, T', and dimensions of geometries g and $g' \in \mathcal{G}'$, we must introduce the notation used in the definition of $VD_{Tc}^{g'}(g)$ (Table 3).

Table 3. Notation used in the definition of the violation degree.

Notation	**Brief description**
S_g	Area of surface g
L_g	Length of curve g
$D_{g,g'}$	Shortest distance between g and g'
$S_{g,g'}$	Area of overlapping between surfaces g and g'
$L_{g,g'}$	Length of overlapping between curves g and g', between curve g and surface g', or between surface g and curve g'
$numCrosses_{g,g'}$	Number of crosses between curves g and g'
L_{min}	Minimum length at the scale of representation
S_{min}	Minimum surface at the scale of representation

The violation degree of an object is relative to the size of the geometry of the object. We use S_g to denote the size (area) of surface g and L_g to denote the size (length) of a curve g.

The interaction between geometries is reflected by their degree of intersection (overlapping or crossing) or separation (distance). For surfaces g and g', $S_{g,g'}$ denotes their area of overlapping, and for curves g and g', $L_{g,g'}$ denotes the length of overlapping. $L_{g,g'}$ also denotes the length of the intersection when g is a surface and g' is a curve, or vice versa. For curves g and g', $numCrosses_{g,g'}$ corresponds to the number of crosses between these curves. This parameter is useful to approximate the violation degree when lines

that cross should be disjoint or touch. For any pair of geometries g and g', $D_{g,g'}$ denotes the minimum distance between them. With some abuse of the notation, $D_{g,g'}$ refers to the external distance between disjoint geometries, but also to the internal distance when a geometry is within another geometry, which corresponds to the distance between their boundaries. Notice that distances apply to any combination of dimensions (e.g., a point with respect to a curve, a curve respect to a surface, and so on).

In addition to measures using the geometries, we need to introduce measures to reflect the area or distance that should exist between geometries but it does not exist. For example, there should be a minimum area of separation between two surfaces that touch but must be disjoint. We denote S_{min} and L_{min} the minimum area of a surface and the minimum length of a curve. These measures depend on the scale of representation of the geometries in the dataset. For example, in a scale 1:5000 we can use $L_{min} = 5$m. However, we note that these parameters are application dependent and we could use different values to work with a higher or lower precision.

The following subsections present the different measures that evaluate the degree of violation of a geometry in a particular dimension with geometries of the same dimension (see Section 4.2 for surfaces and Section 4.3 for curves) and of different dimensions (see Section 4.4 for surfaces with curves and Section 4.5 for points with surfaces and curves). The sections share a common scheme: they contain several tables, each of them representing an expected topological relation between a geometry of study g and another geometry g'. The first column of each table contains the actual relation that holds between g and g', and the second column the formula to compute the violation degree. For example, in Table 4 the first row shows how to compute the violation degree of two geometries when they should be disjoint, but they actually touch each other.

To simplify the presentation, we show how to compute the violation degree of g with respect to one geometry $g' \in \mathcal{G}'$. Recall that, in general, the violation degree of g, $VD_{Tc}(g, D)$, can be computed according to Equation 5. Furthermore, the violation degree of a geometry with respect to another geometry is upper-bounded to value 1.0, which is omitted in the definitions.

We also analyze the case of determining the violation degree with respect to a topological relation that is defined as a disjunction of topological relations defined in Table 1 such as the case of the relation of being internally disconnected which is defined as $\mathsf{Disjoint} \vee \mathsf{Touches}$ (see Section 4.6). Finally, we provide some properties that our inconsistency measure satisfies (see Section 4.7).

4.2. *Violation degree between surfaces*

Table 4 presents the formulas to compute the violation degree of g, when g and geometries in $\mathcal{G}'$ are surfaces. To clarify the rationality of the definitions, we provide examples for some of the entries. We do not provide examples for each definition for considerations of space, but we include cases that reflect the use of the different elements in Table 3.

Example 4.2 Consider the case when surfaces g and $g' \in \mathcal{G}'$ are disjoint and they should touch each other (see Figure 4). The degree of violation is defined in terms of the separation between g and g' denoted by $D_{g,g'}$. Since the violation degree is relative to the area of surface g, the separation (a distance) is transformed to an area by $D_{g,g'} \times L_{min}$, where L_{min} is the minimum length at the scale of representation of the dataset. Indeed, in any of the defined measures, when a distance must be transformed to an area for normalization, it is multiplied by L_{min}. Therefore, the violation degree of g with respect

Table 4. Definition of the violation degree of g when g and g' (a geometry in $\mathcal{G}'$) are surfaces.

Expected relation: Disjoint	
Touches	$\frac{S_{min}}{S_g}$
Overlaps	$\frac{S_{g,g'}+S_{min}}{S_g}$
Within	1
Equals	1
Contains	$\frac{S_{g'}+D_{g,g'}\times L_{min}+S_{min}}{S_g}$
Touch	
Disjoint	$\frac{D_{g,g'}\times L_{min}}{S_g}$
Overlaps	$\frac{S_{g,g'}}{S_g}$
Within	1
Equals	1
Contains	$\frac{S_{g'}+D_{g,g'}\times L_{min}}{S_g}$
Overlap	
Disjoint	$\frac{D_{g,g'}\times L_{min}+S_{min}}{S_g}$
Touches	$\frac{S_{min}}{S_g}$
Within	$\frac{D_{g,g'}\times L_{min}+S_{min}}{S_g}$
Equals	$\frac{S_{min}}{S_g}$
Contains	$\frac{D_{g,g'}\times L_{min}+S_{min}}{S_g}$

Within	
Disjoint	1
Touches	1
Overlaps	$\frac{S_g-S_{g,g'}}{S_g}$
Equals	0
Contains	$\frac{S_g-S_{g'}}{S_g}$
Equal	
Disjoint	1
Touches	1
Overlaps	$\frac{S_g-S_{g,g'}+\lvert S_g-S_{g'}\rvert}{S_g}$
Within	$\frac{\lvert S_g-S_{g'}\rvert}{S_g}$
Contains	$\frac{\lvert S_g-S_{g'}\rvert}{S_g}$
Contains	
Disjoint	$\frac{max(0,S_{g'}-S_g)+D_{g,g'}\times L_{min}+S_{g'}}{S_g}$
Touches	$\frac{max(0,S_{g'}-S_g)+S_{g'}}{S_g}$
Overlaps	$\frac{max(0,S_{g'}-S_g)+S_{g'}-S_{g,g'}}{S_g}$
Within	$\frac{S_{g'}-S_g}{S_g}$
Equals	0

to a surface g' with which it is disjoint and should touch is $\frac{D_{g,g'}\times L_{min}}{S_g}$. □

Example 4.3 Consider the case when a surface g contains but should overlap a surface g' (see Figure 5). The degree of violation is defined in terms of the internal separation of boundaries between g and g' denoted by $D_{g,g'}$ plus the minimum area of overlapping S_{min} at the scale of representation of the dataset. Like the measure described in Example 4.2, this separation is transformed to an area by $D_{g,g'} \times L_{min}$. Finally, the violation degree of g with respect to its conflict with g' is $\frac{D_{g,g'}\times L_{min}+S_{min}}{S_g}$. □

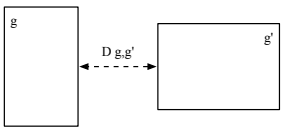


Figure 4. Example of geometries that are disjoint and should touch.

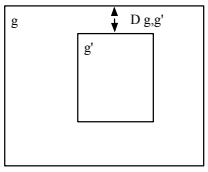


Figure 5. Example of a surface g that contains and should overlap g'.

Example 4.4 Consider the case when a surface g overlaps a surface g' but it should be equal (see Figure 6). The degree of violation is defined in terms of the area of g that does not intersect g' plus the difference in the size of both surfaces. The violation degree of g with respect to its conflict with g' is $\frac{S_g-S_{g,g'}+\lvert S_g-S_{g'}\rvert}{S_g}$. □

Example 4.5 Consider the case when a surface g touches a surface g' but it should contain it (see Figure 7). The degree of violation is defined in terms of whether or not the size of g can contain the size of g' plus the size of g' that has to move inside g. Then, the violation degree of g with respect to its conflict with g' is $\frac{max(S_g, max(0, S_{g'} - S_g) + S_{g'})}{S_g}$. □

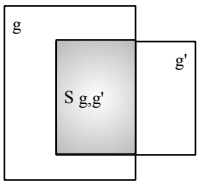


Figure 6. Example of a surface g that overlaps and should be equal to a surface g'.

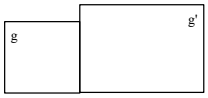


Figure 7. Example of a surface g that touches and should contain a surface g'.

4.3. *Violation degree between curves*

Table 5 presents the violation degree of g, when g and geometries in $\mathcal{G}'$ are curves.

The following examples complement the description of these measures.

Example 4.6 Consider the case when a curve g crosses a curve g' but it should be disjoint (see Figure 8). The degree of violation is defined in terms of a minimum length of a curve L_{min} and the number of times the lines cross $numCrosses_{g,g'}$. The number of times the lines cross defines the number of segments that are part of the conflict. To avoid the computation of the length of each segment, we approximate the violation degree by considering the minimum length that these segments should have. The violation degree of g with respect to its conflict with g' is then $\frac{numCrosses_{g,g'} \times L_{min}}{L_g}$. □

Example 4.7 Consider the case when a curve g should cross a curve g' but instead it is disjoint (see Figure 9). The degree of violation is defined in terms of the distance between both curves $D_{g,g'}$ plus the minimum segment that should cross L_{min}. The violation degree of g with respect to its conflict with g' is then $\frac{D_{g,g'} + L_{min}}{L_g}$. □

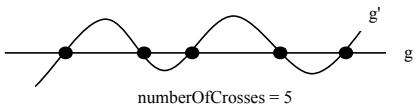


Figure 8. Example of a curve g that crosses and should be disjoint from curve g'.

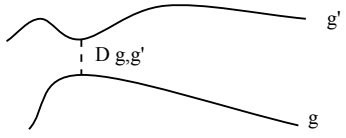


Figure 9. Example of a curve g that is disjoint and should cross a curve g'.

Due to space considerations, we do not illustrate with an example the definition of $VD^{g'}_{Tc}(g)$ for each type of $T \in Tc$. We encourage readers to follow definitions for surfaces, which are similar in spirit to the definitions for curves.

Table 5. Definition of the violation degree of g, when g and g' are curves.

Expected relation: Disjoint	
Touches	$\frac{L_{min}}{L_g}$
Overlaps	$\frac{L_{g,g'}+L_{min}}{L_g}$
Crosses	$\frac{L_{min}\times numCrosses_{g,g'}}{L_g}$
Within	1
Equals	1
Contains	$\frac{L_{g'}+L_{min}}{L_g}$
Touches	
Disjoint	$\frac{D_{g,g'}}{L_g}$
Overlaps	$\frac{L_{g,g'}}{L_g}$
Crosses	$\frac{L_{min}\times numCrosses_{g,g'}}{L_g}$
Within	1
Equals	1
Contains	$\frac{L_{g'}}{L_g}$
Overlaps	
Disjoint	$\frac{D_{g,g'}+L_{min}}{L_g}$
Touches	$\frac{L_{min}}{L_g}$
Crosses	$\frac{L_{min}}{L_g}$
Within	$\frac{L_{min}}{L_g}$
Equals	$\frac{L_{min}}{L_g}$
Contains	$\frac{L_{min}}{L_g}$
Crosses	
Disjoint	$\frac{D_{g,g'}+L_{min}}{L_g}$
Touches	$\frac{L_{min}}{L_g}$
Overlaps	$\frac{L_{g,g'}}{L_g}$
Within	1
Equals	1
Contains	$\frac{L_{g'}}{L_g}$

Within	
Disjoint	1
Touches	1
Crosses	1
Overlaps	$\frac{L_g-L_{g,g'}}{L_g}$
Equals	0
Contains	$\frac{L_g-L_{g'}}{L_g}$
Equals	
Disjoint	1
Touches	1
Crosses	1
Overlaps	$\frac{L_g-L_{g,g'}+\lvert L_g-L_{g'}\rvert}{L_g}$
Within	$\frac{\lvert L_g-L_{g'}\rvert}{L_g}$
Contains	$\frac{\lvert L_g-L_{g'}\rvert}{L_g}$
Contains	
Disjoint	$\frac{max(0,L_{g'}-L_g)+D_{g,g'}+L_{g'}}{L_g}$
Touches	$\frac{max(0,L_{g'}-L_g)+L_{g'}}{L_g}$
Crosses	$\frac{max(0,L_{g'}-L_g)+L_{g'}}{L_g}$
Overlaps	$\frac{max(0,L_{g'}-L_g)+(L_{g'}-L_{g,g'})}{L_g}$
Within	$\frac{L_{g'}-L_g}{L_g}$
Equals	0

4.4. *Violation degree between a surface and curves, and vice versa*

Table 6 presents the violation degree of g when, in the second column, g is a surface and g' is a curve, and in the third column, g is a curve and g' is a surface.

Like above, the following examples explain the rationality of some of these definitions.

Example 4.8 Consider the case when a surface g should be disjoint a curve g' but instead, g' crosses g (see Figure 10). The degree of violation is defined in terms of the length of the segments of g' that intersect g, $L_{g,g'}$, plus the minimum separation between g and g' in terms of a distance L_{min}. We multiply the previous sum by the minimum length L_{min} to make it relative to the size of surface g. The violation degree of g with respect to g' is then $\frac{(L_{g,g'}+L_{min})\times L_{min}}{S_g}$. When taking the same case, but from the point of view of curve g', the violation degree is defined in the same way without considering the transformation of the length of the intersection plus the minimum distance to an area.

Table 6. Definition of the violation degree of g, where g is a surface and g' is a curve, and vice versa.

	Surface point of view	Curve point of view
Expected relation: Disjoint		
Touches	$\frac{S_{min}}{S_g}$	$\frac{L_{min}}{L_g}$
Crosses	$\frac{(L_{g,g'}+L_{min})\times L_{min}}{S_g}$	$\frac{L_{g,g'}+L_{min}}{L_g}$
Contains	$\frac{(D_{g,g'}+L_{g'}+L_{min})\times L_{min}}{S_g}$	N/A
Within	N/A	1
Touches		
Disjoint	$\frac{D_{g,g'}\times L_{min}}{S_g}$	$\frac{min(L_g,D_{g,g'})}{L_g}$
Crosses	$\frac{L_{g,g'}\times L_{min}}{S_g}$	$\frac{L_{g,g'}}{L_g}$
Contains	$\frac{(D_{g,g'}+L_{g'})\times L_{min}}{S_g}$	N/A
Within	N/A	1
Crosses		
Disjoint	$\frac{(D_{g,g'}+L_{min})\times L_{min}}{S_g}$	$\frac{min(L_g,(D_{g,g'}+L_{min}))}{L_g}$
Touches	$\frac{S_{min}}{S_g}$	$\frac{L_{min}}{L_g}$
Contains	$\frac{(D_{g,g'}+L_{min})\times L_{min}}{S_g}$	N/A
Within	N/A	$\frac{D_{g,g'}+L_{min}}{L_g}$
Contains		
Disjoint	$\frac{(D_{g,g'}+L_{g'})\times L_{min}}{S_g}$	N/A
Touches	$\frac{L_{g'}\times L_{min}}{S_g}$	N/A
Crosses	$\frac{(L_{g'}-L_{g,g'})\times L_{min}}{S_g}$	N/A
Within		
Disjoint	N/A	1
Touches	N/A	1
Crosses	N/A	$\frac{L_g-L_{g,g'}}{L_g}$

The violation degree of g' with respect to g is then $\frac{(L_{g,g'}+L_{min})}{L_{g'}}$. □

Example 4.9 Consider the case when a surface g is disjoint but should be crossed by a curve g' (see Figure 11). The degree of violation is defined in terms of the distance $D_{g,g'}$ plus the minimum length of a segment of g' that should cross g. We multiply the previous sum by the minimum length L_{min} to make it relative to the size of surface g. The violation degree of g with respect to g' is then $\frac{(D_{g,g'}+L_{min})\times L_{min}}{S_g}$. Considering the same case from the point of view of the curve g', the violation degree is defined in the same way without considering the transformation of the length of intersection plus the minimum distance to a surface. The violation degree of g' with respect to g is then $\frac{D_{g,g'}+L_{min}}{L_{g'}}$. □

Example 4.10 Consider the case when a surface g is crossed by a curve g' but g should contain g' (see Figure 12). The degree of violation of g is defined in terms of the length of the segment of g' that is not contained in g, multiplied by the minimum length L_{min} to make it relative to the size of surface g. The violation degree of g with respect to g' is then $\frac{(L_{g'}-L_{g,g'})\times L_{min}}{S_g}$. In the same case from the point of view of curve g', the violation degree is defined in the same way without the multiplication of the intersection between

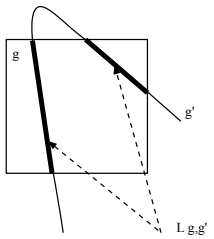


Figure 10. Example of a surface g that should be disjoint to a curve g' but g' crosses g.

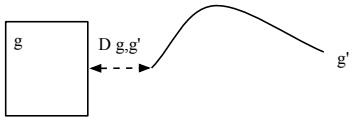


Figure 11. Example of a surface g that is disjoint but should be crossed by a curve g'.

g and g' by L_{min}, since we do not need to transform the intersection to an area. The violation degree of g' with respect to g is then $\frac{L_{g'}-L_{g,g'}}{L_{g'}}$. □

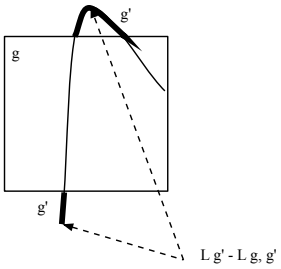


Figure 12. Example of a surface g that is crossed by a curve g' that it should contain.

4.5. *Violation degree between a point and curves or surfaces, and vice versa*

Table 6 shows the violation degree of g, when g is a point and g' is a surface (first part) or g' is a curve (second part). For the first part, the violation degree is defined for the surface and also for the point. Likewise, for the second part, the violation degree is defined for the curve and also for the point.

Table 7. Definition of the violation degree of g, when g is a point, g' is a surface or a curve, and vice versa.

	Surface and point		**Curve and point**	
	Surface	Point	Curve	Point
Disjoint				
Contains	$\frac{(D_{g,g'}+L_{min})\times L_{min}}{S_g}$	N/A	$\frac{D_{g,g'}+L_{min}}{L_g}$	N/A
Within	N/A	1	N/A	1
Contains				
Disjoint	$\frac{D_{g,g'}\times L_{min}}{S_g}$	N/A	$\frac{D_{g,g'}}{L_g}$	N/A
Within				
Disjoint	N/A	1	N/A	1

4.6. *Violation degree for derived relations*

We have defined different measures of the violation degree of spatial objects with respect to the relations defined in Table 1. However, our definition of topological constraints allows the usage of derived relations, defined as a disjunction of topological relations. Therefore, we now define the strategy to determine the violation degree of a geometry g with respect to this type of derived relation.

For a derived relation $T = T_1 \vee \cdots \vee T_n$, with T_i a topological relation in Table 1, the violation degree with respect to T is defined by comparing the relation T' between geometries to its conceptually closest $T_i \in T$ (see Figure 2). This idea is supported by the fact that our measures are in agreement with the semantic distance measure defined over the conceptual neighboring graph (Egenhofer and Al-Taha 1992, Egenhofer and Mark 1995). In fact, when comparing the relation T' between geometries and an expected relation T, semantically close relations have lower degree of violation than relations far apart in the conceptual neighboring graph. The following example illustrates this idea.

Example 4.11 Consider the derived relation IDisjoint defined as the disjunction Touches$\vee$Disjoint. Let us assume that g and g' are geometries that overlap and should be disjoint. In this case, the closest relation between Overlaps and both Touches or Disjoint is Touches. Thus, the violation degree of g is then defined by the definition that compares Overlaps to Touches. □

4.7. *Properties of the inconsistency measure*

The analysis of the properties of the proposed inconsistency measure uses the concept of *culprits*, which is adapted from Martinez *et al.* (2007).

Definition 4.12: Let D be a spatial database instance that includes a collection of geometries $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{G}|}\}$ and $\mathcal{TC}$ be a set of topological constraints defined over D. Also, let g and g' in $\mathcal{G}$ such that g and g' must hold a topological relation $T \in \mathcal{TC}$, but instead they hold a relation T' with $T \neq T'$. Then the pair (g, g') is a *culprit*.

Culprits are pairs of geometries that cause a violation of a topological constraint. Then, let $culprits(D, \mathcal{TC})$ denote the set of culprits in D with respect to $\mathcal{TC}$.

From the definition of $culprits(D, \mathcal{TC})$, it trivially follows that if $D' \subseteq D$ then $Culprits(D', \mathcal{TC}) \subseteq Culprits(D, \mathcal{TC})$, that is, it is monotonic with respect to D.

Proposition 4.13: Let D be a spatial database instance and $\mathcal{TC}$ a set of topological constraints defined over D. Based on the definition of $culprits(D, \mathcal{TC})$, $InC(D, \mathcal{TC})$ holds the following properties:

(i) $InC(D, \mathcal{TC}) = 0$ if and only if $Culprits(D, \mathcal{TC}) = \emptyset$.
(ii) If $D' \subseteq D$, then $InC(D', \mathcal{TC}) \leq InC(D, \mathcal{TC})$.
(iii) If $(g, g') \in Culprits(D, \mathcal{TC})$, then $InC(D \setminus \{g, g'\}, \mathcal{TC}) \leq InC(D, \mathcal{TC})$.

These properties formalize the expected behavior of the consistency of a database: a consistent database cannot contain culprits and it is possible to reduce the inconsistency of a database by removing culprits.

5. Experimental evaluation

5.1. *Evaluation approach*

In this section we present an experimental evaluation of the inconsistency measure presented in this paper and compare it with the other two measures. The first one, named *Violations* in subsequent tables and figures, corresponds to the number of objects in the dataset that incur in at least one violation of a topological constraint (a similar definition was given by Martinez *et al.* (2007) in the context of relational databases). The second one, named *Semantic Distance* (SD), quantifies inconsistencies as the normalized semantic distance in the conceptual graph of topological relations (Papadias *et al.* 1998), such as the one presented in Figure 2. For example, if the expected relationship between two spatial objects is touch but they actually overlap, the inconsistency according with the semantic distance measure is $1/4$[1].

The following experimental evaluation focuses on checking whether or not the measures of the violation degree of objects presented in Section 4 are able to capture the difference of the relation actually held by objects with respect to the topological relation that they should hold. Thus, we assume in the evaluation that for each database instances a set of constraints has been defined that specifies the expected topological relation between every pair of objects. In particular, the set of experiments has been designed to check the following two hypothesis: $\mathcal{H}1$) Global inconsistency: If the modification of an object causes a violation of an integrity constraint, the inconsistency measure of the database increases. $\mathcal{H}2$) Local inconsistency: The larger the violation degree of an object, the larger the inconsistency measure is. Our experimental evaluation shows that our proposed measure validates both hypothesis, whereas the other measures (*Violations* and *Semantic Distance*) validate only hypothesis $\mathcal{H}1$.

5.2. *Datasets*

In our experiments we use several TIGER/Line Shapefiles from the U.S. Census Bureau[1]. These files are known for not containing inconsistencies in the data. Hence, we artificially introduce inconsistencies in the dataset by means of simplification algorithms and other kind of affine transformations available in the GeoTools Toolkit[2]. Specifically, we use several shapefiles containing data from the state of New York, such as *counties* (62 surfaces), primary and secondary roads (12,162 curves), point landmarks (21,389 points), and also the geometry of the state (1 surface). For each layer, we introduce errors to four different percentages of the total number of objects in the layer (5%, 25%, 50%, and 100%), and also with four different levels of modification (*Tiny*, *Small*, *Medium*, and *High*). The modified objects were selected uniformly at random. This produces a 4×4 matrix where upper-left cells show inconsistencies in database instances where we introduced minor modifications in a few percentage of the objects, whereas lower-right cells show inconsistencies in database instances where we introduced more severe modifications to a high percentage of the objects. For our measure (and also for the semantic distance), these matrices show the average violation degree per object in the

[1]This is considering a maximal distance of 4 in the conceptual graph, which is the case in Figure 2. In our experiments we assume a maximal distance of 3, since we are using current spatial SQL languages that exclude CoveredBy and Covers relations.

[1]http://www.census.gov/geo/maps-data/data/tiger.html

[2]GeoTools (http://www.geotools.org) is an open source Java library that provides tools for geospatial data.

Table 8. Results of the test *county* Touches *county*

		Geometries modified			
Tolerance	**Measure**	**5%**	**25%**	**50%**	**100%**
Tiny	**Violations**	9	44	60	62
	SD	0.09140	0.47850	0.77419	0.87634
	VD	5.23E-5	3.36E-4	5.70E-4	6.40E-4
Small	**Violations**	20	55	58	60
	SD	0.13978	0.52150	0.77419	0.86021
	VD	8.28E-4	0.004020	0.00459	0.00497
Medium	**Violations**	11	50	60	61
	SD	0.07527	0.52150	0.77419	0.88709
	VD	0.00381	0.02752	0.04097	0.07456
High	**Violations**	10	41	57	62
	SD	0.06989	0.36559	0.71505	0.90322
	VD	0.00598	0.03172	0.07089	0.12239

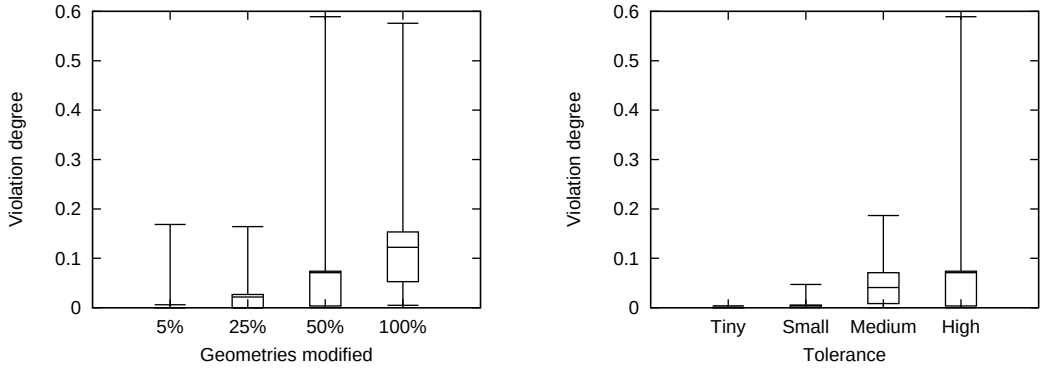
dataset, which is equivalent to the inconsistency measure $InC(D, \mathcal{TC})$ when $\mathcal{TC}$ is the set of topological constraints used to create the inconsistent datasets in the experiments. We aggregate measures per object using averages instead of sums, as described in the paper, in order to introduce a statistical analysis of the results.

5.3. *Inconsistency measures between surfaces*

We first evaluate the measures when the topological constraint involves only surfaces. This is the case of the relationship between counties (i.e., neighboring counties must touch). In this experiment, to introduce inconsistencies we run a simplification algorithm over the set of counties. We control the amount of error introduced in each object by means of the tolerance applied by the simplification algorithm. Specifically, we run a simplification algorithm by Douglas and Peucker (1973) with tolerances from 0.001% (*Tiny*) to 1% (*High*). The 4×4 matrix of inconsistencies is shown in Table 8. Our simplification algorithm introduces two kinds of inconsistencies for pairs of objects that must touch: Overlaps and Disjoint, where the former is more frequent (note that the whole common boundary between adjacent counties have to be simplified in order to make them disjoint). Both kinds of inconsistencies contribute to the semantic distance in the same amount (i.e., both Disjoint and Overlaps are at distance one from Touches in the conceptual graph of topological relationships). Our measure VD uses the measures of the violation degree according to the formulas shown in Table 4: $\frac{S_{g,g'}}{S_g}$, for the case of Overlaps, and $\frac{D_{g,g'} \times L_{min}}{S_g}$ for the case of Disjoint. In this case, we use a value of $L_{min} = 10^{-6}$, which are roughly $0.1m$ in the Equator.

In this experiment, according to hypothesis $\mathcal{H}1$ we expect that the larger the number of simplified objects, the higher the violation of the consistency. This corresponds with an horizontal view of the matrix (i.e., left-to-right). If we focus on a specific row (for example, *high* tolerance), we observe that all the three measures increase with the number of modified objects increases. In Figure 13(a), we present a detailed analysis of this horizontal view for our measure. Specifically, we plot the violation degree of the objects in the dataset when 5%, 25%, 50%, and 100% of the objects were simplified with a high tolerance. This variable, number of *geometries modified*, has been shown statistical significant by means of an Analysis Of Variance (ANOVA)[1].

[1]The statistical significance of the results has been tested in the same way in the following experiments. Hereinafter, we omit this kind of graph in order to improve the legibility of the paper.



(a) Horizontal view (high tolerance) (b) Vertical view (50% of geometries modified)

Figure 13. Detailed view of the test *county* Touches *county*

Table 9. Results of the test *county* Within *state*

		Geometries modified			
Tolerance	**Measure**	**5%**	**25%**	**50%**	**100%**
Tiny	**Violations**	2	8	19	34
	SD	0.01075	0.04301	0.10215	0.1827
	VD	4.13E-6	4.35E-5	1.11E-4	1.81E-4
Small	**Violations**	0	9	17	34
	SD	0	0.04838	0.09139	0.18279
	VD	0	7.94E-4	0.00135	0.00223
Medium	**Violations**	2	4	15	31
	SD	0.01075	0.02150	0.08064	0.16667
	VD	1.62E-4	2.73E-4	0.00178	0.00453
High	**Violations**	1	7	17	29
	SD	0.00537	0.03763	0.09139	0.15591
	VD	1.33E-6	0.00187	0.00260	0.006962

The violation of the consistency should also increase with the tolerance applied by the simplification algorithm ($\mathcal{H}2$). This corresponds with a vertical view of the matrix (i.e., top-to-bottom). Our proposal stands out in this scenario because it is the only measure that captures this expected behavior. This is because our measure is not just proportional to the number of violations, but also to the degree in which objects participate in such violations. Note that the semantic distance remains quite stable in a vertical view because the number of violations also remains stable (and all the violations are of type Disjoint or Overlaps). In Figure 13(b), we plot the violation degree when the tolerance applied by the simplification algorithm was tiny, small, medium, and high. For each group, 50% of the objects in the dataset were simplified.

Next, we validate the same hypothesis with a similar experiment. In this case, we use the geometries of the counties of the New York state, which must be Within the boundary of such state (i.e., we check the topological constraint *county* Within *state*). We also vary the number of modified geometries and the tolerance used by the simplification algorithm. All the introduced violations are of type Overlaps, thus we use the formula: $\frac{S_g - S_{g,g'}}{S_g}$. Table 9 shows the results of this experiment.

The results of this experiment are consistent with the previous ones. We observe that all the measures capture the significant influence of the number of modified objects in the violation of the consistency, whereas the influence of the tolerance applied in the simplification algorithm is only captured by our proposal. We also notice that the values obtained in this experiment are lower. The reason is that we modified a percentage of

the objects in the dataset, but only modifications to counties close to the border of the state produce a violation of the tested constraint. Therefore, when we modify just a few number of objects (up to 25%) results are not significant.

Finally, Table 10 presents a summary of the results of the dual experiment to the previous one: we simplified the geometry of the state of New York with different tolerances and test in which degree the new geometry still contains the counties of such state (i.e., we check the topological constraint *state* Contains *county*). All the introduced violations are of type Overlaps, thus we use the formula: $\frac{max(0,S_{g'}-S_g)+S_{g'}-S_{g,g'}}{S_g}$.

Table 10. Results of the test *state* Contains *county*

	Tolerance			
Measure	**Tiny**	**Small**	**Medium**	**High**
Violations	1	1	1	1
SD	1.0	1.0	1.0	1.0
VD	1.55E-5	2.06E-4	0.00101	0.01005

These results reinforce the importance of our proposal. Both the number of violations and the semantic distance are not useful in this scenario because they return the same value for all levels of simplification: There is one object (the state of New York) that violates the constraint. Its geometry should contain the geometries of the counties, but instead it overlaps with some of them. Note that each of these overlaps contributes in 1/3 to the semantic distance, and thus, if there are at least 3 overlaps, the global measure SD reaches the maximum value 1. Unlike these measures, our proposal obtains larger values for higher tolerances applied during the simplification process.

5.4. *Inconsistency measures between curves*

In order to evaluate our measures for curves, we use a dataset composed of 12,162 primary and secondary roads in the state of New York. Similarly to the previous set of experiments, we introduce some inconsistencies in the dataset by means of transformations to the original objects. We present here an experiment that captures a common inconsistency in real databases of roads: two roads must Touches but they actually are Disjoint or Crosses each other. The Touches relationship is many times necessary in order to navigation algorithms to work. However, when creating cartography (specially with the tools available a few years ago) is easy to make the mistake of not creating the segments of the exact length, but a bit shorter (so they are Disjoint) or larger (so they Crosses). In our original dataset we identified 24,023 pairs of roads that touch each other. The following table presents the behavior of the studied measures when we introduce some modifications in the dataset. Note that for this experiment we cannot simplify the geometries, because simplification algorithms usually keep endpoints unaltered and roads usually touch at endpoints (at least at the endpoint of one of them). Instead, we applied an affine transformation that scales geometries relatively to their centroids. The two kinds of inconsistencies that exist in the modified datasets are disjoint (mainly for roads that we scaled down) and cross (mainly for roads that we scaled up). We used the formulas $\frac{D_{g,g'}}{L_g}$ and $\frac{L_{min} \times numCrosses_{g,g'}}{L_g}$, respectively.

Results in Table 11 show that our measures are also adequate for datasets with curves. Similarly to the case of surfaces, note that whereas all the measures capture the influence of the number of objects with inconsistencies (horizontal view), our proposal is the only one that captures the influence of the amount of inconsistency (vertical view).

Table 11. Results of the test *road* Touches *road*

		Geometries modified			
Scale	**Measure**	**5%**	**25%**	**50%**	**100%**
Tiny	**Violations**	921	3,190	4,603	5,262
	SD	0.03475	0.14874	0.24812	0.30902
	VD	0.00444	0.02353	0.04323	0.07736
Small	**Violations**	956	3,281	4,635	5,262
	SD	0.03626	0.15260	0.24653	0.30902
	VD	0.00711	0.03701	0.06642	0.11083
Medium	**Violations**	936	3,233	4,650	5,267
	SD	0.03604	0.14994	0.24828	0.30913
	VD	0.01864	0.08014	0.13720	0.20513
High	**Violations**	927	3,293	4,603	5,259
	SD	0.03579	0.15507	0.24688	0.30918
	VD	0.02492	0.09802	0.16252	0.24287

Table 12. Results of the test *road* Within *state*

		Geometries modified			
Scale	**Measure**	**5%**	**25%**	**50%**	**100%**
Tiny	**Violations**	16	60	126	243
	SD	4.39E-4	0.00164	0.00345	0.00666
	VD	5.69E-5	2.58E-4	5.50E-4	0.00103
Small	**Violations**	21	70	173	351
	SD	5.75E-4	0.00191	0.00474	0.00962
	VD	2.37E-4	7.96E-4	0.00213	0.00410
Medium	**Violations**	21	99	205	442
	SD	5.75E-4	0.00271	0.00561	0.01211
	VD	3.39E-4	0.00159	0.00323	0.00713
High	**Violations**	46	272	523	1011
	SD	0.00126	0.00745	0.01438	0.02776
	VD	8.45E-4	0.00586	0.01106	0.02147

5.5. *Inconsistency measures between a surface and curves, and vice versa*

To evaluate the measures in this scenario, we design an experiment that combines two of the datasets already described: roads and states. In this case, we introduce inconsistencies in the layer of roads by means of an affine transformation similar to the one used in the previous experiment. The sole difference is that for this experiment we scale the geometries to increase their length, whereas in the previous one we scaled them in order to both increase and decrease their length. This modification produces that some of the roads cross the boundary of the state to which they belong (i.e., we check the topological relation *road* Within *state*). The results of this experiment are summarized in Table 12, and we used the formula: $\frac{L_g - L_{g,g'}}{L_g}$

The results of this experiment in Table 12 are consistent with previous ones and corroborate the adequacy of our proposal to measure both types of inconsistencies described in hypothesis $\mathcal{H}1$ and $\mathcal{H}2$. However, we notice that in this experiment the two other measures also seem capable of distinguishing both types of inconsistencies. In this case, as we increase the scale factor, we also increase the number of objects that violate the topological constraint. Note that when the scale factor is small, just roads close to the boundary may violate the constraint but, as we increase the scale factor, we also increase the number of candidates that may violate the constraint.

Table 13. Results of the test *state* Contains *landmark*

	Tolerance			
Measure	**Tiny**	**Small**	**Medium**	**High**
Violations	1	1	1	1
SD	1.0	1.0	1.0	1.0
VD	1.01E-4	6.85E-4	0.00341	0.01010

5.6. *Inconsistency measures between a point and curves or surfaces, and vice versa*

The experiment designed to show the behavior of the measures involving points is similar to the last one presented in Section 5.3 (*state* Contains *county*), but we use landmark points in the state of New York instead of counties. In addition we apply larger tolerance values in the simplification algorithm. The checked topological constraint is *state* Contains *landmark*, and the violations are mostly of type Disjoint. Thus, we use the formula $\frac{D_{g,g'} \times L_{min}}{S_g}$. Table 13 shows the results of this experiment.

Obviously, the number of violations and the semantic distance are not significant because they relate with the number of objects that violate the constraint, which in this case is the geometry of the state of New York. However, our proposal performs similar to the previous scenarios and it also proves suitable for datasets containing points.

6. Conclusions and future work

We have proposed a new inconsistency measure of spatial datasets that can be used not only to compare the quality of different spatial datasets, but also to quantify how many spatial objects should be modified to restore consistency. Our measure is simple and the experimental evaluation has proved its applicability in many practical scenarios. We also define measures that provide information about the violation degree of each particular object (in relation with the others), which is important to define cleaning procedures that restore the consistency of spatial datasets.

We have proposed measures for the degree of violation of three different types of spatial objects (surfaces, curves, and points) and combinations of them. For each pair of elements of any of these types, we have shown the topological constraints that can be defined over them, and how these constraints may be violated (defining a measure that quantifies such inconsistency). These measures are based on intrinsic properties of the objects (length, area, etc.) and also on properties of the relationship (distance, overlapping area, etc.). The choice of the factors that define the inconsistency was based on a cognitive study by Brisaboa *et al.* (2011).

We have also showed how to aggregate these measures for each object in such a way that we obtain a measure of the inconsistency of the object. We propose the use of the bounded sum, which is simple, efficient, and works well in the majority scenarios. However, we also illustrated with an example a case where we cannot consider violations of the consistency as independent events and thus, the sum over-counts the degree of inconsistency. In these cases, we propose to compute the measure of inconsistency of each object with respect to the geometric union of all the other objects. Nevertheless, the computational cost of the geometric union is high and its use should be decided carefully.

A complete characterization of the cases where the use of the geometric union is unavoidable constitutes our first line of future work. We believe that the degree of overlapping of the objects in the dataset is the key to develop such characterization. We

also plan to study more efficient alternatives for these cases (for example, the usage of simplifications such as minimum bounding rectangles). Finally, the definition of cleaning algorithms (guided by the measures defined in this paper) that allow the consistency restoration of spatial datasets is in our plans too.

Acknowledgement(s)

This work was supported in part by Ministerio de Ciencia e Innovación (PGE and FEDER) [grant TIN2009-14560-C03-02], Xunta de Galicia (FEDER) [2010/17], Agrupación Estratéxica (FEDER) [CN 2012/211], and Fondecyt-Chile 1080138.

References

Baeza-Yates, R.A. and Ribeiro-Neto, B.A., 1999. *Modern Information Retrieval*. Addison Wesley.

Berreti, S., Bimbo, A.D., and Vicario, E., 2000. The Computational Aspect of Retrieval by Spatial Arrangement. *In*: *International Conference on Pattern Recognition*.

Bonnissone, P. and Tong, R., 1985. Reasoning with Uncertainty in Expert Systems. *International Journal of Man and Machine Studies*, 22, 241–250.

Borges, K., Davis, C., and Laender, A., 2002. Integrity Constraints in Spatial Databases. *In*: *Database Integrity: Challenges and Solutions* Ideas Group.

Borges, K., Laender, A., and Davis, C., 1999. Spatial Integrity Constraints in Object Oriented Geographic Data Modeling. *In*: *ACM-GIS*, 1–6.

Bosc, P. and Prade, H., 1997. An Introduction to Fuzzy Set and Possibility Theory Based Approaches to the Treatment of Uncertainty and Imprecision in Datatabase Management Systems. *In*: *Uncertainty Management in Information Systems: From Needs to Solutions* Kluwer Academic Publishers, 285–324.

Bravo, L. and Rodríguez, M.A., 2012. Formalizing and Reasoning about Spatial Semantic Integrity Constraints. *Data & Knowledge Engineering*, 72, 63–82.

Brisaboa, N., Luaces, M., and Rodríguez, M.A., 2011. Cognitive Adequacy of Topological Consistency Measures. *In*: *Advances in Conceptual Modeling. Recent Developments and New Directions - ER 2011 Workshops*, Vol. 6999 of *LNCS* Springer, 241–250.

Douglas, D.H. and Peucker, T.K., 1973. Alogrithms for the reudction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10 (2), 112–122.

Egenhofer, M. and Al-Taha, K., 1992. Reasoning about Gradual Change of Topological Relationships. *In*: *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, LNCS 636* Springer-Verlag, 196–219.

Egenhofer, M. and Franzosa, R., 1991. Point Set Topological Relations. *IJGIS*, 5, 161–174.

Egenhofer, M. and Mark, D., 1995. Modeling Conceptual Neighborhoods of Topological Line-Region Relations. *International Journal of Geographic Information Science*, 9 (5), 555–565.

Egenhofer, M. and Shariff, A., 1998. Metric Details for Natural-Language Spatial Relations. *ACM Transactions on Information Systems*, 16 (4), 295–321.

Egenhofer, M.J. and Dube, M.P., 2009. Topological relations from metric refinements. *In*: *ACM-GIS*, 158–167.

Godoy, F.A. and Rodríguez, M.A., 2004. Defining and Comparing Content Measures of Topological Relations. *GeoInformatica*, 8 (4), 347–371.

Hadzilacos, T. and Tryfona, N., 1992. A Model for Expressing Topological Integrity Constraints in Geographic Databases. *In*: *Spatio-Temporal Reasoning*, Springer LNCS 639, 252–268.

ISO, 2004. *ISO 19125-1:2004 Geographic information – Simple feature access – Part 1: Common architecture.* Technical report, International Organization for Standardization.

Jiang, L., *et al.*, 2009. Measuring and Comparing Effectiveness of Data Quality Techniques. *In*: *Advanced Information Systems Engineering, 21st International Conference, CAiSE 2009*, 171–185.

Lee, J. and Kwan, M.P., 2005. A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments. *International Journal of Geographical Information Science*, 19 (10), 1039–1056.

Martinez, M.V., *et al.*, 2007. How Dirty Is Your Relational Database? An Axiomatic Approach. *In*: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU 2007*, 103–114.

Mäs, S., 2007. Reasoning on Spatial Semantic Integrity Constraints. *In*: *COSIT*, Springer LNCS 4736, 285–302.

Motro, A. and Rakov, I., 1996. Estimating the Quality of Data in Relational Databases. *In*: *IQ* MIT, 94–106.

OGC, 2011. *OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.* Technical report, Open Geospatial Consortium, Inc.

Ordonez, C., García-García, J., and Chen, Z., 2007. Measuring referential integrity in distributed databases. *In*: *Proc. of CIMS Workshop - 16th ACM-CIKM* ACM.

Papadias, D., Mamoulis, N., and Delis, V., 1998. Algorithms for Querying Spatial Structure. *In*: *VLDB Conference*, 546–557.

Parson, S., 1996. Current Approaches to Handling Imperfect Information in Data and Knowledge Bases. *IEEE Transactions on Knowledge and Data Engineering*, 8 (3), 353–371.

Plümer, L. and Gröger, G., 1997. Achieving Integrity Constraints in Geographic Information Systemsd. *GeoInformatica*, 1 (4), 345–367.

Randell, D., Cui, Z., and Cohn, A., 1992a. A Spatial Logic based on Regions and Connection. *In*: *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning* Morgan Kaufmann, 165–176.

Randell, D., Cui, Z., and Cohn, A., 1992b. A Spatial Logic Based on Regions and Connection. *In*: B. Nebel, C. Rich and W. Swarthout, eds. *Principles of Knowledge Representation and Reasoning* Morgan Kaufmann, 165–176.

Rodríguez, M.A., Bertossi, L.E., and Marileo, M.C., 2013. Consistent Query Answering under Spatial Semantic Constraints. *Information Systems*, 38 (2), 244–263.

Rodríguez, M.A., *et al.*, 2010. Measuring consistency with respect to topological dependency constraints. *In*: *18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*, 182–191.

Servigne, S., *et al.*, 2000. A Methodology for Spatial Consistency Improvement of Geographic Databases. *GeoInformatica*, 4 (1), 7–34.

Zlatanova, S., Rahman, A.A., and Shi, W., 2004. Topological models and frameworks for

3D spatial objects. *Computers & Geosciences*, 30 (4), 419 – 428.