Dipartimento di Informatica
Università degli Studi di Verona

Rapporto di ricerca
Research report
85/2011
October 2011

Integrating Multi-Accuracy Spatial Data

Alberto Belussi

alberto.belussi@univr.it

Sara Migliorini

sara.migliorini@univr.it

Questo rapporto è disponibile su Web all'indirizzo:
This report is available on the web at the address:
http://www.di.univr.it/report

Abstract

In recent years the integration of spatial data coming from different sources has become a crucial issue for many geographical applications, in particular in the process of building and maintaining a Spatial Data Infrastructure (SDI). In such context new methodologies are necessary in order to acquire and update spatial datasets by collecting new measurements from different sources. The traditional approach implemented in GIS systems for updating spatial data does not usually consider the accuracy of these data, but just replaces the old geometries with the new ones. The application of such approach in the case of an SDI, where continuous and incremental updates occur, will lead very soon to an inconsistent spatial dataset with respect to spatial relations and relative distances among objects. In this report we address this problem and we propose a framework for representing multi-accuracy spatial databases, based on a statistical representation of the objects geometry, together with a method for the incremental and consistent update of the database objects, that applies a customized version of the Kalman filter. Moreover, in the framework we consider also the spatial relations among objects, since they represent a particular kind of observation, that could be derived from geometries or be observed independently in the real world. Therefore, also spatial relations among objects coming from different sources need to be compared and we show that they are necessary in order to obtain a correct result in objects geometry integration.

Keywords: spatial data integration, multi-accuracy spatial data, statistical update, Kalman filter.

1 Introduction

During the last years the attention of geographical applications towards the spatial data integration problem has rapidly increased. For instance, many geographical national o regional agencies, in particular in the European Union, are facing the challenge of integrating in common Spatial Data Infrastructures (SDIs) spatial data coming from different sources and acquired using different technologies and instruments. Therefore, in the GIS community there is a need for new data integration methods to consolidate huge amount of spatial data belonging to different thematic layers. In particular, those methods have to be able to integrate different observations regarding the same specific and identified geographical object (or set of objects) or about different objects among which a particular relation holds. In doing this, such methods have to consider the metadata describing the quality of both the datasets to be integrated and the resultant one, and this is an important issues for the following reasons.

Spatial objects representing geographical features are inherently uncertain because the measurements needed to survey the shape, extension and position of an object with the maximal accuracy are too expensive, or because the maximal accuracy is not necessary to satisfy the application requirements. Therefore, a certain amount of error in the representation of a spatial object always exists. In literature [Sch99, TN02b, Hop08] the term *accuracy* is considered as a measure of how closely the recorded values represent their true values, while *uncertainty* is a statistical estimate of the accuracy of a value and thus it is modeled using probability theory. However, the importance of uncertainty is perceived in different ways by the different communities that are working in the GIS field.

Considering in particular the vector representation of spatial data (i.e. spatial datasets are sets of geometries including points, polylines and polygons specified by a list of coordinates in a reference space) we can observe that: *computer scientists* working with GIS tend to perceive absolute coordinates as the primary data concerning location of geographical objects and to consider geometric coordinates as deterministic values. The measurements from which these coordinates were obtained are seen as unnecessary data once the absolute point locations have been determined; no record is kept about the measurements from which they are derived. In this perspective each relative geometry measure (e.g. distance, angle, etc) and all the other information (e.g. spatial relationships between objects) can be derived from absolute coordinates. On the contrary, *surveyors* typically perceive the measurements concerning geographical objects and the relative object distances as being the primary data, while the computed coordinates are treated as random variables. Coordinate values are seen simply as a view of the data: the one that best fits the measurements at that time. The accuracy of relative geometry is in practice higher than the absolute accuracy; therefore,

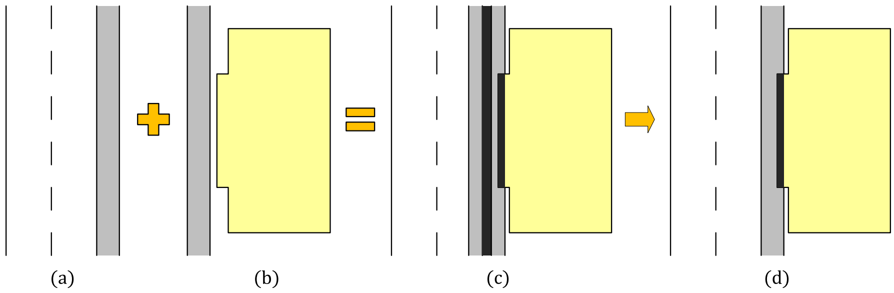


Figure 1: An example of integration that does not consider the accuracies of the objects to be integrated, but simply replace old objects with new ones.

absolute coordinates and relative measures are not equivalent as computer scientists often believe.

Although it is possible to store measurements, rather than derived coordinates, into a database and calculate the coordinates as required using all the stored measurement information, this operation is computationally intensive and hence it is not practical. As a consequence spatial databases usually store only derived coordinates without any information about their accuracy. However, some accuracy information have to be attached to spatial data, since the derivation of coordinates from observations is a unique but not reversible operation [GGA04a]. Moreover, information about accuracy of spatial data should be used in every operation involving these uncertain data; in particular, it is fundamental for integrating new observations coming from different sources, or for correctly interpreting the result of a query. We also observe that the result of integrating spatial data coming from different sources is a dataset containing multi-accuracy spatial data and hence it is crucial that accuracy becomes a part of spatial data representation at single object granularity.

Example 1. Fig. 1 illustrates a typical problem that occurs when an integration is performed simply by replacing older or less accurate objects with newer or more accurate ones. In particular, Fig. 1(a) and Fig. 1(b) represent two source databases that have one object in common: a sidewalk that is depicted as the light gray polygon in both databases. Each database contains also an additional object, namely a road and a building, respectively. Moreover, a disjoint condition is defined between the sidewalk and the building in the second database and we suppose to know that it is the existing relation between them in the real world. Now suppose that the first database has a higher absolute metric accuracy than the second one, so in the integrated database the resulting sidewalk is the one of Fig 1(a), while the other one is discarded. Finally, the building is simply added in the resulting database without modifying its geometry and without any consideration about its accuracy and its relations with other objects. The resulting database is reported in Fig. 1(d): the building overlaps the sidewalk, violating the disjoint condition defined in the second source database. This is a consequence of the relative positions between the two geometries representing the sidewalk in the two source databases as shown in Fig. 1(c).

The contribution of this report is articulated into two points: firstly, in Sec. 3 we define a methodology for calculating and representing the accuracy of spatial data; secondly, in Sec. 4 we propose an integration procedure, based on the Kalman filter, that considers the accuracy of both the coordinates of the source databases and the topological relations defined among database objects, producing an integrated database with updated accuracies. Finally, some properties of the proposed integration procedure are discussed in Sec. 5.

Before presenting the proposed framework for handling multi accuracy spatial data, the following section illustrates some previous works related to our proposal.

2 Related Work

The need to consider the accuracy of spatial data is widely recognized in literature. In particular, in [NRB03, BLR+04, LBR+07] Bhanu et al. propose a probability-based method for modeling and indexing uncertain spatial data. In this model each object is represented by a probability density function and the authors discuss how to perform spatial database operations in presence of uncertainty. In particular, in [NRB03] they present a method for performing the probabilistic spatial join operation, which, given two uncertain datasets, finds all pairs of polygons whose probability of overlap is larger than a given threshold. In [BLR+04, LBR+07] Bhanu et al. present a different indexing structure, called Optimized Gaussian Mixture Hierarchy (OGMH) that supports both uncertain/certain queries on uncertain/certain data, in particular they consider the k nearest neighbors (k-NN) search operation. The proposed model allows the representation of multi-accuracy spatial databases, because the uncertainty of an object is described by associating to each vertex of its extent a probability density function. Therefore, an object can be intended as a d-dimensional random variable and the similarity between two objects is given by the probability that the two corresponding random variables are the same.

Another model for representing uncertainty in spatial database is proposed by Tossebro et al. in [TN02a, TN02b, TN02c, TN03, TN04]. In [TN02b] the authors propose a representation of spatial data through uncertain points, uncertain lines and uncertain regions. The basic idea is that all uncertain objects, regardless of their type, are known to be within a certain crisp region, it may also be known where an object is most likely to be. So they define the concepts of *core* and *support*: each object is represented by two regions, one inside the other, the innermost region is the area in which the object is certain to be, it is called core and it is the area of greatest probability; the outermost region is the area in which the object may be, it is called support and in this area the probability of the object is above 0. Moreover, it is known that the object is not outside the outermost region. In [TN03] this model is refined in order to reduce the storage space required and to simplify the computation of the core and support regions. In [TN02c] the authors extends its model with some constructs for representing also temporal uncertainty into a spatial database. Finally, in [TN04] the model is completed with the representation of topological relationships between uncertain spatial objects, since they cannot be directly inferred from the object representations.

Unfortunately none of this work deals with the integration process, they propose a more or less formal model for representing uncertainty and eventually they concentrate on query operations. Conflation techniques [Saa88] have been widely used for integrating two vector spatial databases. These methods essentially involves two phases: (1) corresponding features in the two source datasets are recognised through the identification of matching control points, (2) the two source datasets are aligned using rubber-sheeting transformations based upon the identified matching control points. These phases are repeated iteratively, with further control points being identified as the data sources are brought into alignment. However, conflation techniques typically align the dataset with lower accuracy to the more accurate one, called target dataset. The positional information related to the control points within the less accurate dataset is ignored, assuming that the target dataset is correct. In this way, corresponding features in the two datasets are aligned but in a sub-optimal manner. Moreover, any updated quality information are provided for the adjusted dataset.

A more sophisticated approach to the integration problem has to take into account the accuracies of both source datasets in order to produce a more accurate integrated database, as done in [GGA04b, HKH06, HK08, Hop08]. These approaches use techniques based on weighted least squares method to obtain the best fit between the source datasets. The advantage of such an approach is that resultant positions are optimized taking into account all the available information, including the positional accuracy of points in both datasets. Moreover, updated quality parameters are generated for each point, enabling detailed quality reporting of the resultant dataset. In [HKH06, HK08, Hop08] the authors consider also the problem of preserving topological relations between objects by representing them as inequality observations that could be included in the least squares method. These inequalities are obtained by combining the collinearity and equality constraints with three other conditions: a point being on the left/right of a line segment, a point being at a minimum/maximum distance from another point or from a line segment.

In [SB97] the authors discuss how to use the Kalman filter into a static context for sequentially improving the best least squares estimate as soon as new observations are integrated. The key concept above the use of the Kalman filter is the idea of updating the solution: the new estimate is expressed as the linear combination of the previous one and the new observations, in a recursive manner, so that it is not required to store the previous integrated observations. In [Alt93] the author uses the Kalman filter approach to estimate the coordinate positions of atoms within a molecule. He assumes a static structure and he does not introduce any time-dependent model of change.

These solutions for updating spatial data rely on measure with known accuracy, therefore they are not directly applicable to existing spatial databases containing only coordinate values. A method has to be defined for determining the accuracy of these coordinates from the commonly available information.

3 Representing Multi-Accuracy Spatial Databases

A multi-accuracy database is a spatial database in which objects are characterized by different accuracy parameters, in the extreme case each single point in the database can have a different accuracy. In this section we present an abstract data model for representing multi-accuracy spatial databases, called *MACS database*.

Spatial information can be classified into two major groups: *metric observations* and *logic observations*. Metric observations represent quantitative properties of spatial objects, in particular their position and extension. These observations are subject to uncertainty and have to be treated with a statistical approach in order to express their different accuracies. Logic observations describe qualitative properties of spatial objects, like spatial relations or shape characteristics. This kind of observations represents certain information, namely they can be only known or unknown and so they are treated with a logical approach. In geographic applications the most important category of spatial relations is the set of topological ones. Many models for this kind of relations have been proposed in literature, starting from the well known 9-intersection model of Egenhofer et al. [EF91]. In this paper we assume that metric observations and topological relations are stored inside a MACS database and they are considered jointly during the update phase, which integrates new metric or logic observations with the existing ones, or the integration phase where another MACS database is integrated with the current one.

3.1 Representing Metric Observations

A MACS database is constituted by a set of objects, called *features*, adopting the terminology of the ISO TC 211 International Standards for geographical information and the Open GeoSpatial Consortium. A feature represents a real geographic entity and has a fundamental property which is the geometry describing its extension, shape and position on the Earth surface. In a MACS database each *real position* P is represented as a pair of random variables (x_P, y_P) (we consider 2D datasets) and its accuracy information is expressed by the joint probability density function:

$$f_P(x_P, y_P) : E^2 \rightarrow [0, 1] \tag{1}$$

This function describes where the position P could be located; its type depends on the survey process and can vary considerably. In this work we assume that random variables representing real positions have a Gaussian distribution, since statistically this is the distribution obtained by any experimental process.

Following this approach, for each position P to be stored in the database, it should be necessary to store its joint probability density function by means of a set of parameters that approximate the function $f_P(x_P, y_P)$. This set of parameters could be very large, moreover visualizing complex probability density functions or using them in query processing could be very difficult. Thus, a synthetic description of $f_P(x_P, y_P)$ has to be defined. Considering the context of geographical applications of recent years, where very few information about spatial accuracy is available, we propose to adopt the following representation of absolute positions.

Definition 1 (Soft Absolute Position)**.** The absolute position of a point P with probability density function $f_P(x_P, y_P)$, is represented by a position index and a dispersion index. The *position index* of P, also called *representative point* and denoted by $\underline{\mathrm{P}}$, is the point (μ_x, μ_y), where μ_x and μ_y are the averages of x and y with respect to $f_P(x_P, y_P)$. The *dispersion index* of P represents the dispersion of the probability around $\underline{\mathrm{P}}$ and is given by the variance-covariance matrix of the x and y variables.

$$C_\sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \tag{2}$$

□

In many real situations (as the building of a national SDI) the only available metadata describing the metric quality of coordinates are an error estimate e for the absolute position, that is the maximum granted error between the real coordinates and the measurement, and a validity percentage of that error *FR(e)*, that is the percentage of cases that have to satisfy this error, for each surveyed area. In [CCHF+98] the authors illustrate how variance of coordinates can be calculated from this information using the circular error formula; in this paper we adopt their approach, as shown in Eq. 3. Since there is no reason for considering different the variance of x from the variance of y, we can suppose that:

$$\sigma^2_{x_P} = \sigma^2_{y_P} = \sigma^2 = \frac{-e^2}{2 \cdot \log(1 - F_R(e))} \tag{3}$$

Moreover, the covariance between the x and y coordinates of the same absolute position P is set to zero, assuming that the two components are mutually independent.

Given the variance and covariance of a position P, the correlation between different positions can be estimated by modeling the covariance between them as a function inversely proportional to their distance. In this way the correlation is greater for near points and it decreases as distance increases. Let us consider a MACS database containing only two positions $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$, the matrix C_{DB} containing the variance and covariance of P and Q is defined as:

$$C_{DB} = \begin{bmatrix} \sigma^2_{x_P} & \sigma_{x_P,y_P} & \sigma_{x_P,x_Q} & \sigma_{x_P,y_Q} \\ \sigma_{y_P,x_P} & \sigma^2_{y_P} & \sigma_{y_P,x_Q} & \sigma_{y_P,y_Q} \\ \sigma_{x_Q,x_P} & \sigma_{x_Q,y_P} & \sigma^2_{x_Q} & \sigma_{x_Q,y_Q} \\ \sigma_{y_Q,x_P} & \sigma_{y_Q,y_P} & \sigma_{y_Q,x_Q} & \sigma^2_{y_Q} \end{bmatrix} \tag{4}$$

In order to simplify the model and reduce the number of unknown parameters in the matrix, we formulate the following hypothesis.

Definition 2 (Independence hypothesis)**.** Considering surveyed spatial data, the following hypothesis can be reasonable in applications that deal with them:

1. The measurement methods currently available allow one to assume that the x and y coordinates of a position P are mutually independent, so the covariance between the x and y components of P can be set to zero: $\sigma_{x_P,y_P} = 0$

2. Accordingly with the previous assumption, the x coordinate of a position P does not influence the y coordinate of any other point Q, so: $\sigma_{x_P,y_Q} = 0$

3. The correlation between the x coordinate of a position P and the x coordinate of an other position Q is equal to the correlation between the y coordinate of P and the y coordinate of Q: $\sigma_{x_P,x_Q} = \sigma_{y_P,y_Q}$

4. The variance of the x coordinate of a position P coincides with the variance of its y coordinate, since the value of absolute accuracy are the same: $\sigma^2_{x_P} = \sigma^2_{y_P} = \sigma^2_P$ □

Applying the hypothesis contained in Def. 2, the variance-covariance matrix C_{DB} for two positions P and Q can be rewritten as follows:

$$C_{DB} = \begin{bmatrix} \sigma_P^2 & 0 & c_{PQ} & 0 \\ 0 & \sigma_P^2 & 0 & c_{PQ} \\ c_{PQ} & 0 & \sigma_Q^2 & 0 \\ 0 & c_{PQ} & 0 & \sigma_Q^2 \end{bmatrix} \tag{5}$$

where $c_{PQ} = \sigma_{x_P,x_Q} = \sigma_{y_P,y_Q}$ is the correlation between the positions P and Q, while $\sigma_P^2 = \sigma_{x_P}^2 = \sigma_{y_P}^2$ and $\sigma_Q^2 = \sigma_{x_Q}^2 = \sigma_{y_Q}^2$.

The remaining unknown parameter is only c_{PQ}. In order to obtain an estimation of this parameter we propose the following approach: c_{PQ} represents somehow the "attraction" that P exerts on Q and vice versa, thus we can estimate it by considering the accuracy of the relative distance among points of the map. Indeed, this is another piece of metadata that is often available for surveyed spatial datasets, since the accuracy of the relative distance among the surveyed objects is usually higher than the one derivable from the accuracy of the absolute coordinates of points. Now supposing that $\sigma_{d_{PQ}}^2$ is the variance of the relative distance between the two positions P and Q, that can be calculated using Eq. 3, where e is replaced with the maximum granted error of the relative distance between absolute positions and $F_R(e)$ with its percentage of validity, c_{PQ} can be calculated as shown in the following lemma.

Lemma 1 (Covariance estimation)**.** *Given the variance $\sigma_{d_{PQ}}^2$ of the relative distance between the two points P and Q and the variance of their coordinates σ_P^2 and σ_Q^2, the covariance $\sigma_{x_P,x_Q} = \sigma_{y_P,y_Q} = c_{PQ}$ can be calculated as follows:*

$$c_{PQ} = \frac{\sigma_P^2 + \sigma_Q^2 - \sigma_{d_{PQ}}^2}{2} \tag{6}$$

Proof. - (sketch) Eq. 6 is obtained by applying the variance propagation law to the random variable d_{PQ}, representing the distance $\overline{PQ}$, and the vector of random variables $\overline{v} = (x_P\ y_P\ x_Q\ y_Q)$, representing the coordinates of the points P and Q. The relation $d_{PQ} = g(\overline{v})$ exists, where g is the well-known function that calculates the distance between two points. Notice that g is a non-linear function, but it can be easily linearized as $d_{PQ} \simeq J\overline{v}$, where J is the Jacobian (the matrix containing the partial derivatives of g with respect each component of $\overline{v}$). According to the variance propagation law: $\sigma_{d_{PQ}} = J \cdot C_v \cdot J^T$ and from here, considering as C_v the matrix in Eq. 5, we obtain the thesis. $\square$

Let us notice that there is a connection between the accuracy of absolute positions of two points and the accuracy of their relative distance. For example, if two points P and Q have an absolute accuracy corresponding to a circular error of e_P and e_Q respectively with a percentage of 95%, than their relative distance will be affected at most by an error of $e_P + e_Q$ in the 95% of the cases. Moreover, we also remark in the following observation that, in the context of real spatial data integration, only positive values of covariance are acceptable in order to preserve relative distances among points.

Observation 1 (Positive covariance constraint)**.** *In order to preserve the relative distance between two position P and Q during the integration and update process presented in the following sections, the covariance value c_{PQ} between P and Q has to be positive (greater than zero), namely from Eq. 6:*

$$\sigma^2_{d_{PQ}} < \sigma^2_P + \sigma^2_Q \tag{7}$$

It follows that every time a value of $\sigma^2_{d_{PQ}}$ greater than this limit is obtained from Eq. 3, it must be substituted with the value $\sigma^2_P + \sigma^2_Q$. □

The reasoning illustrated above regards only two positions, but its extension to the network of all points contained in a database is straightforward. In particular, this procedure must be applied to all possible pair of positions in the database, altogether there are $m = \binom{n}{2}$ pairs of positions, where n is the total number of positions. It is easy to show that the procedure applied considering all the n positions is equivalent to the application of the procedure to the m pairs of positions in input.

Note that the variance-covariance matrix is symmetric; therefore, only its upper (or lower) triangle has to be stored, halving the required space. Anyway, it is clear that the dimension of the matrix for a real database grows rapidly and its complete storage into a database becomes difficult but not impracticable. Indeed for each position P with two coordinates (x_P, y_P), two variance and $(n-1)/2$ covariance values have to be stored. Thus for a database of n points, requiring to store $2n$ real numbers, we need to store additionally $2n+n(n-1)/2$ real numbers for the variance-covariance matrix. However, some optimizations can be adopted, in particular, notice that:

- At the beginning the variance-covariance matrix can be calculated starting from two metadata describing the metric quality of the dataset: $(e, FR(e))$, the error the absolute coordinates and its percentage of validity, and $(e_d, FR(e_d))$ the same for the relative distance among points, as shown in the above presented reasoning.

- After the integration with other datasets the variance-covariance matrix will of course change with respect to the calculated one, but in many cases significant variations are local and so only the portions of the matrix that have been substantially changed with respect to the calculated one (e.g. where variations are above a fixed threshold) have to be stored.

- At a given moment we can also partition the reference space of the database in regions having homogeneous metric accuracy and store for each of these areas two metadata describing their metric quality: $(avg(e), FR(avg(e)))$ of the absolute coordinates and $(avg(e_d), FR(avg(e_d)))$ of the relative distance among points.

Given the notion of absolute position, a geometric object is defined as follows.

Definition 3 (Object or feature)**.** An object O is defined as: $O = \langle ID, Geo, CL\rangle$ where:

- *ID* is an integer representing an unique identifier for the object;
- *Geo* is the geometry of the object. It is composed of: (i) the set of absolute positions $Geo.pos = \{P_1, \ldots, P_n\}$ describing the geometry and its uncertainty, (ii) the type of geometry $Geo.type \in \{point, curve, surface\}$ and (iii) the representative geometry $Geo.rep = \{\mu_{x_1}, \mu_{y_1}, \ldots, \mu_{x_n}, \mu_{y_n}\}$ which is the point, polyline or polygon used during object visualisation and querying. In order to handle the case in which only spatial relations among objects are represented (see next section), without any information about geometries, the empty value for *Geo* is admitted. It is denoted as $\emptyset_{geo}$ and we suppose that $\emptyset_{geo}.pos = \emptyset_{geo}.rep = \emptyset$ and $\emptyset_{geo}.type = null$.
- *CL* is the thematic class to which the object belongs, for example Building or Road. □

Notice that on the geometry of each object the following constraints hold: if $Geo.type = point$, then $|Geo.pos| = |Geo.rep| = 1$, if $Geo.type = curve$, then $|Geo.pos| = |Geo.rep| > 1$, if $Geo.type = surface$, then $|Geo.pos| = |Geo.rep| > 2$.

3.2 Representing Logic Observations

For representing geographical information, another kind of observation is necessary, namely the spatial relations among the objects of a dataset. Several types of spatial relations can be considered; in this paper we focus on topological relations, since they have been deeply studied in literature starting from the paper of Egenhofer [EF91] and they are available in every current GIS product and also open source software, like the well known Java APIs such as JTS Topology Suite.[1].

[1] www.vividsolutions.com/jts/jtshome.htm

Many different models for the definition of topological relations have been proposed starting from the well-known *9-intersection model* defined in [EF91, EF95]. In particular, since the objects we are considering have geometries of different types (point, curve and surface), we adopt the set of topological relations defined by Clementini et al. in [CFvO93]. This is a complete set of mutually exclusive topological relations, namely a set of topological relations in which for each pair of objects there is one and only one possible relation. In the 9-intersection model, the geometry of each object A is represented by 3 point-sets: its interior A°, its exterior A^-, and its boundary ∂A. The definition of binary topological relations between two spatial objects A and B is based on the 9 possible intersections of each object component. Thus, a topological relation $R(A, B)$ can be represented as a 3×3-matrix, called *9-intersection matrix*, defined as:

$$R(A,B) = \begin{bmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{bmatrix}$$

Considering the value empty ($\oslash$) or not empty ($\neg\oslash$) for each intersection, many relations can be distinguished between surfaces, curves and points. In [CFvO93], this model has been extended by considering for each 9-intersection its dimension (i.e., 0 for points, 1 for curves and 2 for surfaces), giving raise to the *extended 9-intersection model*. Since the number of such relations is quite high, a partition of the extended 9-intersection matrices has been defined, grouping together similar matrices and assigning a name to each group. The result is the definition of the following set of binary, mutually exclusive topological relations: {*Disjoint*, *Touch*, *In*, *Contain*, *Overlap*, *Cross*, *Equal*}. We also consider the relations *CoveredBy* and *Covers*, since they are specializations of *In* and *Contains* for which a specific treatment is necessary during the integration process. The reference set of topological relations considered here is: R_{topo} = {*Disjoint*, *Touch*, *In*, *CoveredBy*, *Contains*, *Covers*, *Cross*, *Overlap*}.

The semantics of topological relations in R_{topo} is provided in Table 1. The last column presents for each topological relation the pattern grouping all the corresponding 9-intersection matrices. Notice that the dimension of the intersection is needed only to discriminate between *Cross* and *Overlap* in the case of pairs of curves; in all the other cases, dimension is not required. The boundary of a geometry is defined as follows: a surface boundary is the ring defining its border, the boundary of a curve is composed of its end points and the point boundary is empty.

In current spatial databases topological relations between objects are usually derived from their geometries. However, in a MACS database absolute positions, composing the objects geometries, are *soft data*, namely they are uncertain. As a consequence, from absolute positions only *soft topo-*

Relation Name	**Relation Definition**	**Geometry type (S: surface, C: curve, P: point)**	**Corresponding patterns of the 9-int. matrix**
disjoint (d)	$A \cap B = \emptyset$	S/S, C/C, S/C, C/S	$FFT-FFT-TTT$
		S/P, C/P	$FFT-FFT-TFT$
		P/S, P/C	$FFT-FFF-TTT$
		P/P	$FFT-FFF-TFT$
touch (t)	$(A^\circ \cap B^\circ = \emptyset) \wedge (A \cap B) \neq \emptyset)$	S/S	$FFT-FTT-TTT$
		C/C	$F*T-*T*-T*T$ $F*T-T**-T*T$ $FTT-***-T*T$
		S/C	$FFT-T**-**T$ $FFT-FTT-T*T$
		C/S	$FT*-F**-T*T$ $FFT-FT*-TTT$
		S/P, C/P	$FFT-TFT-FFT$
		P/C, P/S	$FTF-FFF-TTT$
in (i)	$(A \cap B^\circ = A) \wedge (A^\circ \cap B^\circ) \neq \emptyset)$	S/S, C/C, C/S	$TFF-TFF-TTT$
		P/S, P/C	$TFF-FFF-TTT$
coveredBy (b)	$(A \cap B = A) \wedge (A^\circ \cap B^\circ) \neq \emptyset) \wedge$ $(A \cap B^\circ \neq A)$	S/S, C/C	$TFF-TTF-TTT$
		C/S	$T*F-*TF-TTT$
contains (c)	$(A \cap B^\circ = B) \wedge (A^\circ \cap B^\circ) \neq \emptyset)$	S/S, C/C, S/C	$TTT-FFT-FFT$
		S/P, C/P	$TFT-FFT-FFT$
covers (v)	$(A \cap B = B) \wedge (A^\circ \cap B^\circ) \neq \emptyset) \wedge$ $(A^\circ \cap B \neq B)$	S/S, C/C	$TTT-FTT-FFT$
		S/C	$T*T-FTT-FFT$ $T*T-TFT-FFT$ $T*T-TTT-FFT$
equal (e)	$A = B$	S/S, C/C	$TFF-FTF-FFT$
		P/P	$TFF-FFF-FFT$
cross (r)	$dim(A^\circ \cap B^\circ) =$ $(max(dim(A^\circ), dim(B^\circ)) - 1) \wedge$ $(A \cap B) \neq A \wedge (A \cap B) \neq B$	C/S	$TTT-***-TTT$
		S/C	$T*T-T*T-T*T$
		C/C	$0*T-***-T*T$
overlap (o)	$dim(A^\circ) = dim(B^\circ) =$ $dim(A^\circ \cap B^\circ) \wedge$ $(A \cap B) \neq A \wedge (A \cap B) \neq B$	S/S	$TTT-TTT-TTT$
		C/C	$1*T-***-T*T$

Legend: *The pattern is a string "$c_{1,1}c_{1,2}c_{1,3} - c_{2,1}c_{2,2}c_{2,3} - c_{3,1}c_{3,2}c_{3,3}$", where element $c_{i,j}$ corresponds to cell (i,j) in the 9-intersection matrix. If $c_{i,j} = *$ then this position is not relevant in defining the topological relation, if $c_{i,j} = F/T$ means that the intersection is (or is not) empty, $c_{i,j} \in \{0,1,2\}$ means that the intersection has the specified dimension. Finally, $dim(g)$ computes the dimension of the geometry g.*

Table 1: Definition of the reference set of topological relations between two objects A and B.

logical relations can be derived, namely topological relations that are not precisely defined.

Claim 1. *We claim that also topological relations can be considered as observations useful for representing spatial information. This claim has two important consequences: (i) observed topological relations among objects of a dataset have to be stored independently with respect to objects geometries; (ii) observed topological relations have to be integrated with objects geometries resolving possible inconsistency.* □

Moreover, topological relations cannot be considered data subject to measurement error, since they cannot be measured like the width of a building, they can only be true, false or unknown. Therefore, we will call them *hard data*, to distinguished them from the absolute positions that are *soft data*, as explained before. The uncertainty of the knowledge about the topological relation existing between two objects can be represented by a disjunction of topological relations, that we know might exists between them. If we cannot exclude any relations, then the disjunction is composed of all relations of the considered reference set.

Definition 4 (Hard Topological Relation)**.** Given a complete set of mutually exclusive topological relations R_{topo}, an instance of topological relation is defined as: $\langle O_1, R, O_2 \rangle$ where: O_1, O_2 are objects and $R \in 2^{R_{topo}}$ is the set of topological relations that might exist between O_1 and O_2 (e.g. $\{$*Disjoint*$\}$, $\{$*In*, *Equal*$\}$, $\{$*Touch*, *In*, *Overlap*$\}$, etc.). In particular, sets with more than one relation represent disjunction of topological relations between O_1 and O_2. The set R containing all the topological relationships, called universal relation and denoted with R_U, represents the situation in which the topological relation between O_1 and O_2 is unknown. □

Consequently as the regards to topological relations three situations may occur: (i) if $|R| = 1$ then the relation is known, (ii) if $R = R_U$ then the relation is unknown, and (iii) if $|R| > 1 \wedge R \neq R_U$ then the relation is unknown and could be only one of the relations $r \in R$. In the following, where there is no ambiguity, a hard topological relation will be denoted simply as topological relation.

Even if topological relations cannot be derived from absolute positions, we have to impose a *coherence constraint* between hard and soft topological relations. Given two objects A and B the soft topological relation r_{soft} that exists between them can be computed by considering as geometries their representatives (see Def. 3). For obtaining an effective integration between soft and hard data, r_{soft} has to be compatible with the hard topological relation R explicitly stored, i.e. it must be that: $r_{soft} \in R$.

We can notice that the number of hard topological relations to be stored in a MACS database is large, indeed if the database contains n objects, the total number of hard topological relations to be stored is $n \times (n-1)$, because one topological relation has to be defined between each pair of objects. This could be a large number in real databases, thus some optimizations can be applied in order to reduce the amount of information that have to be stored. The idea is to represent hard topological relations among objects using soft topological relations when possible and store them explicitly only when they are completely or partially unknown (i.e. $1 < |R| \leq |R_U|$).

First of all, we introduce the notion of *support* for a position P ($Supp_P(\alpha)$) as the region around $\underline{P}$ where a given quantity $\alpha < 1$ of the probability to find the position P is located. The support of P visualizes the dispersion index around the representative $\underline{P}$. The form of this region depends on the variance and covariance of P and it is in general an ellipse around the representative $\underline{P}$. According to the independence hypothesis (see Def. 2) $\sigma^2_{x_P} = \sigma^2_{y_P} = \sigma_P$ and $\sigma_{x_P y_P} = 0$, so the support for a position in a MACS database is a circle with radius $2\sigma^2_P$ for $\alpha \simeq 0.95$. Given the notion of support for a position P, an index of maximum dispersion α_M can be defined which represents the quantity of probability that has to be considered during the computation of the support for each position in the database. Therefore, any point outside $Supp_P(\alpha_M)$ cannot be considered an eligible position for P. We now extend the concept of support to the geometry of an object.

Definition 5 (Object support estimation)**.** Given an object $O = \langle ID, Geo, CL\rangle$ the support of O with respect to the maximum dispersion index α_M (denoted by $Supp(O, \alpha_M)$) can be approximated by considering the smallest buffer region[2] that contains the support of all its defining positions *O.Geo.pos*. The real position of an object cannot be outside its support. □

Condition on objects support	**Soft top. relation**	**Stored hard top. relation**	**Hard top. relation**
$Supp(A, \alpha_M) \cap Supp(B, \alpha_M) = \emptyset$	$A\ dj\ B$	-	$\langle A, \{dj\}, B\rangle$
$Supp(A, \alpha_M) \cap Supp(B, \alpha_M) \neq \emptyset$	$A\ r\ B$	-	$\langle A, \{r\}, B\rangle$
$Supp(A, \alpha_M) \cap Supp(B, \alpha_M) \neq \emptyset$	$A\ r_i\ B$	$\langle A, \{r_1, .., r_i, .., r_k\}, B\rangle$	$\langle A, \{r_1, ..., r_k\}, B\rangle$
$Supp(A, \alpha_M) \cap Supp(B, \alpha_M) \neq \emptyset$	$A\ r_i\ B$	$\langle A, R_U, B\rangle$	$\langle A, R_U, B\rangle$

Table 2: Possible cases in the representation of the hard topological relations between two objects A and B (dj = disjoint).

Thanks to the object support, only topological relations between pairs of objects $\langle O_1, O_2\rangle$ that interact (i.e. whose supports are not disjoint) have to be explicitly stored. Given two objects whose supports are disjoint the only possible topological relation between them is the disjoint one.

[2]The buffer operation is a well-known operation available in GIS systems that, given a geometry g and a ray r, computes the region representing the set of points having a distance less or equal to r from g.

In practical cases, the topological relation between two features is known rather than unknown, so given the coherence constraint previously mentioned, we can decide to store only topological relations that contain more than one element and derive the others from the representatives of the objects. Thus, given a pair of objects $\langle O_1, O_2 \rangle$ the possible cases are shown in Table 2.

Given the definition of soft and hard data a MACS database is defined as follows.

Definition 6 (MACS database)**.** A Multi ACcuracy Spatial database (MACS database) is a 6-tuple: $DB_{macs} = (DB, C_{DB}, TY, OBJ, REL, \alpha_M, Supp_{DB})$ where:

- DB is a set of position index (i.e., 2D points coordinates) of the absolute positions contained in the MACS database. For each position index $\underline{\mathrm{P}}$ of a position P we store the following tuple: $\langle ID_P, x_P, y_P \rangle$, where ID_P is the identifier of P, and $\underline{\mathrm{P}} = (x_P, y_P)$.
- C_{DB} is the matrix of dispersion indexes (variance and covariance of coordinates) of DB; we discuss below the problem of storing C_{DB}.
- TY is a set of available feature classes for the objects.
- OBJ is a set of objects $\langle ID, Geo, CL \rangle$ (see Def. 3) belonging to the classes of TY and whose geometry is described through the absolute positions in DB and C_{DB}.
- REL is a set of hard topological relations, which are explicitly stored, since they are not derivable from soft topological relations.

- α_M is the maximum dispersion index and Note that if two objects has intersecting geometries, then they must share some positions representing their common intersections points (for surfaces this constraint is referred to the boundary) □

We propose three different methods for storing C_{DB} that corresponds to three different states of the database:

1. Initially the matrix can be generated starting from two metadata describing the metric quality of the dataset: $(e, FR(e))$ of the absolute positions and $(e_d, FR(e_d))$ of the relative distance among positions, by applying the procedure shown in Sec. 3.1. Therefore, only the tuple $(\sigma_P^2, \sigma_{PQ}, Supp_{DB})$ can be stored, to represent the fact that the variance σ_P^2 and the covariance σ_{PQ} are valid inside the region $Supp_{DB}$.
2. After the integration with another database we may need to store some values of C_{DB}, in this case we store only the values that have a significant difference with respect to σ_P^2 and σ_{PQ}, we call this matrix C_{DB}^{δ}; in this case we store: $(C_{DB}^{\delta}, (\sigma_P^2, \sigma_{PQ}, Supp_{DB}))$.

3. Finally, after the integration another situation can be obtained, where in a region, say R_1, some values of variance and covariance are valid and in another region, say R_2, other values have to be applied; in this last case we store: $(C^{\delta}_{DB}, (\sigma^2_{P_1}, \sigma_{PQ_1}, R_1), (\sigma^2_{P_2}, \sigma_{PQ_2}, R_2))$.

Example 2 (Example of MACS database)**.** Let us consider the database presented in Fig. 1(a), denoted here as DB^1_{macs}. Supposing that for DB^1_{macs} the error e for the absolute position is 0.8 meters with a percentage of validity of 95%, and the error e_d for the relative distance is 0.6 meters with a percentage of 95%, while its maximum dispersion index α_M has value 0.75 and the region representing its support is briefly indicated as *supp*. The representation of this MACS database is reported below. Let us notice that with $DB(id)$ we denote the elements of the vector DB (position index) related to the position with identifier id; similarly, with $C_{DB}(id)$ we denote the elements (variance and covariances) of the C_{DB} matrix related to the position with identifier id.

- $DB^1_{macs}.DB = \{\langle id_{001}, 2456, 9783\rangle, \dots \langle id_{023}, 2456, 7684\rangle, \dots\}$
- $DB^1_{macs}.C_{DB} = (0.25, 0.18, Supp_{DB})$
- $DB^1_{macs}.TY = \{Road, Sidewalk\}$
- $DB^1_{macs}.OBJ = \{\langle obj_1, obj_1.Geo, Road\rangle, \langle obj_2, obj_2.Geo, Sidewalk\rangle\}$
  - $obj_1.Geo.pos = \{\langle DB(id_{001}), C_{DB}(id_{001})\rangle, \dots, \langle DB(id_{023}), C_{DB}(id_{023})\rangle, \dots\}$
  - $obj_1.Geo.type = surface$
  - $obj_1.Geo.rep = \{2456, 9783, \dots, 2456, 7684\}$
- $DB^1_{macs}.REL = \{\langle obj_1, \{Touch, Disjoint\}, obj_2\rangle\}$
- $DB^1_{macs}.\alpha_M = 0.75$
- $DB^1_{macs}.Supp_{DB} = Supp_{DB}$

3.3 MACS database accuracy estimators

In order to evaluate the overall accuracy of a MACS database the following indexes can be defined. In particular, we introduce an index of metric accuracy and an index of certainty for logic observations. We choose to give an estimation of certainty for logic observations, instead of uncertainty, in order to have an index with the same behaviour of the metric accuracy.

Given a position P inside a MACS database DB_{macs} the metric accuracy of its absolute position is defined as the inverse of its variance σ^2_P[3]. Thus

[3]Notice that according to Eq. 5 the variance of the position coordinates is the same and, as we will see in next sections, remain the same also after the integration procedure.

the metric accuracy of the position P, denoted as $acc_M(P)$, is defined as:

$$acc_M(P) = \frac{1}{\sigma_P^2} \tag{8}$$

The average global accuracy estimation of a MACS database DB_{macs} concerning the metric observations can be computed as

$$acc_M(DB_{macs}) = \frac{\sum_{P_i \in DB_{macs}.DB} acc_M(P_i)}{|DB_{macs}.DB|} \tag{9}$$

Moreover, we propose an index for the estimation of the certainty of the logic observations in a MACS database; in particular, given the set of topological relations R_n between two objects O_1 and O_2 (R_n can contains one or more relations), the certainty of this information is estimated as:

$$acc_R(R_n) = \frac{|R_U| - |R_n|}{|R_n| \cdot (|R_U| - 1)}$$

Considering the reference set of topological relations proposed in Sec. 3.2, we obtain:

$$acc_R(R_n) = \frac{7 - |R_n|}{6 \cdot |R_n|} \tag{10}$$

Therefore, the certainty is the highest when $|R_n| = 1$, namely when the relation is known, and it is the lowest when $R_n = R_U$, namely when the relation is unknown.

The average global certainty estimation of a MACS database DB_{macs} concerning the logic observation can be computed as follows:

$$acc_R(DB_{macs}) = \frac{|DB_{macs}.OBJ|^2 - |DB_{macs}.REL| + \sum_{r_i \in DB_{macs}.REL} acc(r_i)}{|DB_{macs}.OBJ|^2} \tag{11}$$

Each known topological relation (i.e. not explicitly stored in REL) has a unit certainty value, so the first term $|DB_{macs}.OBJ|^2 - |DB_{macs}.REL|$ computes the overall certainty of all known relations. To this value the certainty of all unknown topological relations is added ($\sum_{r_i \in DB_{macs}.REL} acc_R(r_i)$). This global value is then normalised with respect to the total number of possible topological relations ($|DB_{macs}.OBJ|^2$), so that the certainty is the highest when all the relations are known and decreases when more relations are unknown.

4 Integrating Multi-Accuracy Spatial Databases

This section deals with the problem of integrating two existing MACS databases. Different situations can occur as shown in Table 3, since the integrating databases can be completely different or can share absolute positions, and/or objects, and/or relations. More specifically, different application scenarios can be represented by the integration of two MACS database: (i) the integration of two size comparable spatial databases describing different geographic themes but sharing a large part of territory (cases A.* in the table). (ii) The integration of two databases describing the same geographic features but on adjacent regions (cases A.* in the table). (iii) The integration of a massive spatial database with some new soft or hard observation about known positions or objects (cases B.* in the table). (iv) The update of the geometries of some known objects in a reference dataset (cases B.* in the table).

The integration of two MACS databases produces as result a new MACS database. In order to classify all the situations that is necessary to handle, we first introduce the general operations needed to integrate two MACS databases defining its component tasks and then we describe each of them separately.

Definition 7 (MACS database integration)**.** Given two MACS databases $DB^1_{macs} = (DB_1, C_{DB_1}, TY_1, OBJ_1, REL_1, \alpha_M, Supp_{DB_1})$ and $DB^2_{macs} = (DB_2, C_{DB_2}, TY_2, OBJ_2, REL_2, \alpha_M, Supp_{DB_2})$ their integration produces a new MACS database $DB^3_{macs} = (DB_3, C_{DB_3}, TY_3, OBJ_3, REL_3, \alpha_M, Supp_{DB_3})$ whose components can be obtained by applying different operations to the components of DB^1_{macs} and DB^2_{macs}, depending on the interaction that exists between them, as reported in Table 3, in particular:

$$DB^3_{macs} = DB^1_{macs} \oplus DB^2_{macs}$$

where:

- $DB_3 = metricPosIntegration(DB_1, DB_2, C_{DB_1}, C_{DB_2})$
- $C_{DB_3} = metricVarIntegration(C_{DB_1}, C_{DB_2})$
- $TY_3 = TY_1 \oplus_{ty} TY_2$
- $OBJ_3 = OBJ_1 \oplus_{obj} OBJ_2$
- $REL_3 = logicRelIntegration(ext(OBJ_1, REL_1), ext(OBJ_2, REL_2))$

The function $ext(OBJ, REL)$ returns the set of all valid relations or relation disjunctions that either are stored in REL or can be derived as soft relations from OBJ geometries. □

Notice that, in Table 3 some combinations are not admissible and are not shown, since the following conditions holds:

$$OBJ_1.ID \cap OBJ_2.ID \neq \emptyset \implies ext(REL_1, OBJ_1) \cap ext(REL_2, OBJ_2) \neq \emptyset$$
$$OBJ_1.ID \cap OBJ_2.ID \neq \emptyset \implies TY_1 \cap TY_2 \neq \emptyset$$

Description of the cases	$\langle \boldsymbol{TY_\cap}, \boldsymbol{OBJ_\cap}, \boldsymbol{DB_\cap}, \boldsymbol{REL_\cap} \rangle$
A. Integration of two independent databases having comparable number of objects and positions	
A.0 - Nothing in common (**no adjustments of objects geometries**)	$\langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$
A.1 - Some classes in common, but no objects and points (**no adjustments of objects geometries**)	$\langle \neg\emptyset, \emptyset, \emptyset, \emptyset \rangle$
A.2 - Some points in common, but no classes, objects and relations (**adjustments of interfering objects geometries**)	$\langle \emptyset, \emptyset, \neg\emptyset, \emptyset \rangle$
A.3 - Some classes and points in common, but no objects (**adjustments of interfering objects geometry**)	$\langle \neg\emptyset, \emptyset, \neg\emptyset, \emptyset \rangle$
A.4 - Some classes, objects and relations in common, but no points (**objects update by geometry replacement and relation integration**)	$\langle \neg\emptyset, \neg\emptyset, \emptyset, \neg\emptyset \rangle$
A.5 - Some classes, objects, points and relations in common (**update by geometry modification and relation integration**)	$\langle \neg\emptyset, \neg\emptyset, \neg\emptyset, \neg\emptyset \rangle$
B. Update of a reference databases $\boldsymbol{DB_m^1}$ with new metric and/or logic observations represented in $\boldsymbol{DB_m^2}$	
B.1 - Some classes and points in common, but no objects ($OBJ_2 = \emptyset$) (**adjustments of some positions**)	$\langle \neg\emptyset, \emptyset, \neg\emptyset, \emptyset \rangle$
B.2 - Some classes and points in common, but no objects ($OBJ_2 \neq \emptyset$) (**new objects insertion**)	$\langle \neg\emptyset, \emptyset, \neg\emptyset, \emptyset \rangle$
B.3 - Some classes, objects and relations in common, but no points ($DB_2 \neq \emptyset$) (**objects update by geometry replacement**)	$\langle \neg\emptyset, OID_2.ID, \emptyset, \neg\emptyset \rangle$
B.4 - Some classes, objects and relations in common, but no points ($DB_2 = \emptyset$) (**objects update by relations integration**)	$\langle \neg\emptyset, OID_2.ID, \emptyset, \neg\emptyset \rangle$
B.5 - Some classes, objects, points and relations in common (**update by geometry modification and relations integration**)	$\langle \neg\emptyset, OID_2.ID, \neg\emptyset, \neg\emptyset \rangle$

Table 3: Possible cases in the integration of two MACS databases. In the second column the tuple $\langle TY_\cap, OBJ_\cap, DB_\cap, REL_\cap \rangle$ represents the intersections $\langle TY_1 \cap TY_2$, $OBJ_1.ID \cap OBJ_2.ID$, $DB_1.ID \cap DB_2.ID$, $ext(REL_1, OBJ_1) \cap ext(REL_2, OBJ_2) \rangle$.

The preliminary operation that is necessary in order to integrate two spatial databases is the identification of common classes, objects and positions. The more the databases are decoupled and come from independent sources, the more this operation is tough. Many works were presented in literature dealing with this important issue, denoted as *schema integration* and *features (point) matching*. In this paper, we suppose that the class, object and position matching has already been solved, since we want to focus on the impact of the spatial accuracy in the integration process based on object geometries. Thus, we suppose that common objects in the two integrating databases share the same *ID* and the same is valid for common positions.

The simplest integration tasks are those regarding classes and objects.

Indeed, the integration of the classes produces simply their union: $TY_1 \oplus_{ty} TY_2 = TY_1 \cup TY_2$, while the integration of the objects is obtained as follows:

$$
\begin{aligned}
& OBJ_1 \oplus_{obj} OBJ_2 = \\
& \{o \mid (o \in OBJ_1 \wedge o.ID \notin OBJ_2.ID) \vee (o \in OBJ_2 \wedge o.ID \notin OBJ_1.ID)\} \cup \\
& \{objPosIntegration(o_1, o_2) \mid o_1 \in OBJ_1 \wedge o_2 \in OBJ_2 \wedge o_1.ID = o_2.ID\}
\end{aligned} \tag{12}
$$

where $objPosIntegration(o_1, o_2)$ is the procedure that identifies which positions have to be integrated and stored in the final database DB^3_{macs} as representatives for the object with identifier $ID = o_1.ID = o_2.ID$. This choice can be done by considering the object surveying date, namely by keeping the positions of the most recent object, even its non matching positions, and discarding instead the non matching positions of the other older object. Otherwise a direct decision of the user that supervises the integration process is necessary.

The next subsections are organized as follows, first the integration of positions (metric observations) is considered in Sec. 4.1; in particular, a statistical method for computing the functions *metricPosIntegration*$(DB_1, DB_2, C_{DB_1}, C_{DB_2})$ and *metricVarIntegration*(C_{DB_1}, C_{DB_2}) is presented. In Sec. 4.2 we concentrate on the problem of integrating topological relations (logic observations); in particular, a method for computing the function *logicRelIntegration*(R_1, R_2) is illustrated. Finally, in Sec. 4.3 we treat the problem of maintaining the consistency between metric and logic observations on the integrated database.

4.1 Integrating Metric Observations

This section presents in detail a method for integrating metric observations contained into two MACS databases. This method is denoted here as $metricPosIntegration_{kalman}$ $(DB_1, DB_2, C_{DB_1}, C_{DB_2})$, where DB_1 and DB_2 are the set of position indexes contained in the two databases, while C_{DB_1} and C_{DB_2} are the corresponding dispersion index matrices. This method is based on the application of the Kalman filter to the vectors of coordinates V_{DB_1}, V_{DB_2}, containing the representative of the positions that have to be integrated, and the matrix of the variance-covariance estimates C'_{DB_1} and C'_{DB_2} for these positions.

The use of the Kalman filter for performing the integration has the following important advantage. The least squares-based methods are able to provide the solution that best fit all the information contained in the source datasets; however, the integration cannot always be performed in one time, but it can be necessary or convenient to perform sequential integrations in order to obtain the final result. For instance, this approach is unavoidable when there are more different sources to integrate or when the size of the considered area requires to perform multiple integration steps, each one on a different sub-area. As stated in [SB97] the Kalman filter can be applied for updating the least squares estimate as new integration are performed, in a recursive manner so that it is not necessary to store the previously integrated observations. Even if the Kalman filter has been designed to work with dynamic systems in which the estimate depends on both the new observations and the time change, that filter can also be applied in a static context, as during the integration of different datasets. In particular, given the current estimate $\hat{x}_{k|k}$, the Kalman filter normally provides the updated solution $\hat{x}_{k+1|k+1}$ into two steps: a *prediction* phase that projects forward (in time) the current state, providing a priori estimate $\hat{x}_{k+1|k}$ based only on the current estimate, and a *correction* phase that corrects the a priori estimate based on the new measurements. In a static system the state does not change in time, so the prediction phase is not necessary: the a priori estimate $\hat{x}_{k+1|k}$ corresponds to the current estimate $\hat{x}_{k|k}$.

Notice that, in order to be effectively integrate two databases, they should share an area, otherwise there is no possibility to define a real correlation between them and no adjustment propagation is possible. Similarly, when a new object has to be integrated inside a preexisting database, some information about its nearest objects has to be provided for correctly positioning it and adjusting dependent objects. However, the proposed method is able to deal with all the cases presented in Table 3, in particular for cases $A.0$, $A.1$ and $B.4$ the following integration functions can be applied.

Observation 2 (Metric integration with no common objects (positions))**.** *Considering cases $A.0$, $A.1$ and $B.4$ of Table 3, the following integration functions can be applied:*

$$metricPosIntegration_{union}(DB_1, DB_2, C_{DB_1}, C_{DB_2}) = [DB_1 \ DB_2]$$

$$metricVarIntegration_{union}(C_{DB_1}, C_{DB_2}) = \begin{bmatrix} C_{DB_1} & C_{zero} \\ C_{zero}^T & C_{DB_2} \end{bmatrix}$$

where the matrix C_{zero} contains only zeros.

Proof. This result is due to Obs. 1 and the hypothesis in Def. 2, since we have no information about the relative distance among objects of the two integrate databases. □

In other words, the covariance σ_{PQ} between pair of positions P and Q, where $P \in DB_1 \wedge P \notin DB_2$ and $Q \in DB_2 \wedge Q \notin DB_1$, is set to zero, as no information is available about their correlation.

In all the other cases, the vectors V_{DB_1}, V_{DB_2} and the matrices C'_{DB_1} and C'_{DB_2} are built in different ways, according to the considered scenario (see Table 3), as show in the following observation .

Observation 3 (Initialization of vectors and matrices for the application of the Kalman filter)**.** *Given two sets of position indexes DB_1, DB_2 and the corresponding dispersion indexes C_{DB_1}, C_{DB_2}, the vectors V_{DB_1}, V_{DB_2} and the corresponding variance-covariance matrices C'_{DB_1}, C'_{DB_2} are build as follows:*

- *cases A.2, A.3, A.5 and B.1, B.5: first we drop from each DB_i ($i \in \{1, 2\}$) the positions that are not contained in any object geometry of OBJ_3 (see Eq. (12))*

$$
\begin{aligned}
V_{DB_1} =& [DB_1 \setminus_{ID} DB_2 \;\; DB_1 \cap_{ID} DB_2 \;\; DB_2 \setminus_{ID} DB_1] \\
V_{DB_2} =& [DB_1 \setminus_{ID} DB_2 \;\; DB_2 \cap_{ID} DB_1 \;\; DB_2 \setminus_{ID} DB_1] \\
C'_{DB_1} =& \begin{bmatrix} \Pi_{1-2,1-2}(C_{DB_1}) & \Pi_{1-2,1\cap 2}(C_{DB_1}) & C_{zero} \\ \Pi_{1\cap 2,1-2}(C_{DB_1}) & \Pi_{1\cap 2,1\cap 2}(C_{DB_1}) & C_{zero} \\ C_{zero} & C_{zero} & C_{\infty} \end{bmatrix} \\
C'_{DB_2} =& \begin{bmatrix} C_{\infty} & C_{zero} & C_{zero} \\ C_{zero} & \Pi_{1\cap 2,1\cap 2}(C_{DB_2}) & \Pi_{1\cap 2,2-1}(C_{DB_2}) \\ C_{zero} & \Pi_{2-1,1\cap 2}(C_{DB_2}) & \Pi_{2-1,2-1}(C_{DB_2}) \end{bmatrix}
\end{aligned}
$$

- *case A.4 and B.2, B.3: DB_2 contains some new positions that do not exist in DB_1 or that have to be completely replaced with the corresponding positions in DB_1. We suppose that DB_2 contains also some information about the accuracy for the relative distance between its positions and some positions in DB_1.*

$$
\begin{aligned}
V_{DB_1} =& [DB_1 \setminus_{ID} DB_2 \;\; DB_1 \cap_{ID} DB_2 \;\; DB_2 \setminus_{ID} DB_1] \\
V_{DB_2} =& [DB_1 \setminus_{ID} DB_2 \;\; DB_2 \cap_{ID} DB_1 \;\; DB_2 \setminus_{ID} DB_1] \\
C'_{DB_1} =& \begin{bmatrix} \Pi_{1-2,1-2}(C_{DB_1}) & C_{zero} & C_{zero} \\ C_{zero} & C_{\infty} & C_{zero} \\ C_{zero} & C_{zero} & C_{\infty} \end{bmatrix} \\
C'_{DB_2} =& \begin{bmatrix} C_{\infty} & \Delta(C_{zero}) & \Delta(C_{zero}) \\ \Delta(C_{zero}) & \Pi_{1\cap 2,1\cap 2}(C_{DB_2}) & \Pi_{1\cap 2,2-1}(C_{DB_2}) \\ \Delta(C_{zero}) & \Pi_{2-1,1\cap 2}(C_{DB_2}) & \Pi_{2-1,2-1}(C_{DB_2}) \end{bmatrix}
\end{aligned}
$$

where $[a\ b\ c]$ represents the vector concatenation, $DB_i \setminus_{ID} DB_j = \{p \mid p \in DB_i \wedge p.ID \notin DB_j.ID\}$, $DB_i \cap_{ID} DB_j = \{p \mid p \in DB_i \wedge p.ID \in DB_j.ID\}$ and $\Pi_{a,b}(C)$ computes the matrix by keeping only the elements $c_{i,j} \in C$ where $i \in a$ and $j \in b$. a(b) can be "$1-2$", which means the row (columns) of positions $p \in DB_1 \setminus_{ID} DB_2$, or "$1 \cap 2$", which means the row (columns) of positions $p \in DB_1 \cap_{ID} DB_2$. Finally, C_{∞} is the matrix containing very high variance values on the main diagonal and zero elsewhere, and $\Delta_{a,b}(C_{zero})$ is a matrix containing the covariance between positions i and j, when known from relative distance measures, or zero otherwise.

Proof. This result is obtained for the first cases by considering that: (i) the two databases have to be represented together and for the non-shared objects we only have one pair of coordinates, thus we simulate to have another pair of coordinates in the other database, equal to the original one, but with very low accuracy; (ii) for the shared objects we have instead two pairs of coordinates with different accuracy and we can populate the matrices accordingly. For the second cases we can observe that DB_2 contains some new points that are not present in DB_1 or that have to be replaced with the ones contained in DB_1. Therefore, each common position contained in DB_1 becomes very inaccurate with respect to the one contained in DB_2 and so its variance is replaced with very high values. Moreover, between some positions in DB_2 and DB_1 some information about the accuracy of relative distance might be known, so this information is eventually inserted into the matrix C'_{DB_2} (this is indicated by the use of the Δ operator). □

Now the application of the Kalman's filter is straightforward.

Method 1 (Position Integration (Kalman's filter))**.** Given the vectors V_{DB_1}, V_{DB_2} and the matrix C'_{DB_1}, C'_{DB_2} the Kalman's filter is applied as follows:

$$V_{DB_3} = V_{DB_1} + K(V_{DB_2} - A \cdot V_{DB_1})$$

K is named *Kalman* or *Gain matrix* and it represents the correction applied to the measurements contained in V_{DB_1} due to the presence of the measurements in V_{DB_2}:

$$K = C'_{DB_1} \cdot (C'_{DB_1} + C'_{DB_2})^{-1} \tag{13}$$

A is the *design matrix* which defines the relation between the observations and the parameters; in this paper we consider only direct measurements and so it can be omitted.

$$V_{DB_3} = V_{DB_1} + K \cdot (V_{DB_2} - V_{DB_1}) \tag{14}$$

□

From V_{DB_3} we can easily obtain DB_3 which represents the result of the function $metricPosIntegration_{kalman}(DB_1, DB_2, C_{DB_1}, C_{DB_2})$.

The filter allows not only to update the coordinates of the position indexes, but also to estimate the accuracy of the resulting database, that is to update the variance-covariance matrix as follows:

$$C_{DB_3} = (I - K) \cdot C'_{DB_1} \cdot (I - K)^T + K \cdot C'_{DB_2} \cdot K^T \tag{15}$$

C_{DB_3} is the result of the function $metricVarIntegration(C_{DB_1}, C_{DB_2})$.

It is clear that in real situation the least squares-based methods cannot be applied to an entire database, in particular for the costs of inverting the involved matrices. In Sec. 3 a concept of threshold for covariance values has been defined, so that covariance values have to be stored only between points that really interact, and only for those points it is reasonable to propagate the integration effects. In the same manner, given the two source datasets, a selection on the database positions can be made, considering during the integration process only the positions which are mutually correlated to the new integrated ones.

4.2 Integrating Logic Observations

This section discusses the problem of integrating logic observations contained into two distinct MACS databases DB^1_{macs}, DB^2_{macs}. In particular, referring to Def. 7, we define a method for computing the function $\mathit{logicRelIntegration}(REL_1, OBJ_1, REL_2, OBJ_2)$.

In order to integrate two set of relations, the significant cases require that the databases share at least one object. Anyway, the proposed method is able to handle any possible cases; indeed, different operations are necessary according to the rate of sharing objects. In particular, if no objects are shared the known relations are all preserved, while the new relations between objects of OBJ_1 and objects of OBJ_2 have to be declared unknown. Actually, considering the support of these objects some more precise relations can be derived by computing the relations among their support as shown in the following observation.

Observation 4 (Objects relations from supports relations)**.** *Given two sets of topological relations R_1 and R_2 among two sets of objects O_1 and O_2 respectively, where $O_1.ID \cap O_2.ID = \emptyset$, the following function can be defined for representing the knowledge about the topological relations existing among the objects of $O_1 \cup O_2$. It is obtained by considering the relations between objects supports:*

$$\mathit{topFromSupp}(R_1, O_1, R_2, O_2) = \{\langle o_1, o_2, r_x\rangle \mid (o_1, o_2) \in O_1 \times O_2 \wedge r_x \in R_{topo}\}$$

where r_x is defined as in Fig. 2.

Proof. Considering Fig. 2 and starting from the first conditional block we can observe that, if the objects supports are disjoint, then for the support definition (Def. 5) the two objects are disjoint. If they have intersecting support, o_1 is a surface and the o_2 support is inside o_1 without touching o_1 boundary, then no points of o_2 can have a position that is outside o_1, thus o_2 *in* o_2. The third conditional block shows a situation that is the inverse of the previous one. Finally, in the last conditional block we say that, if two surfaces has intersecting supports excluding their support boundary, they certainly have intersecting interiors, thus the existing relation between them can be only one of the following relations: *in*, *contains*, *covers*, *coveredBy*, *equal* or *overlap*. □

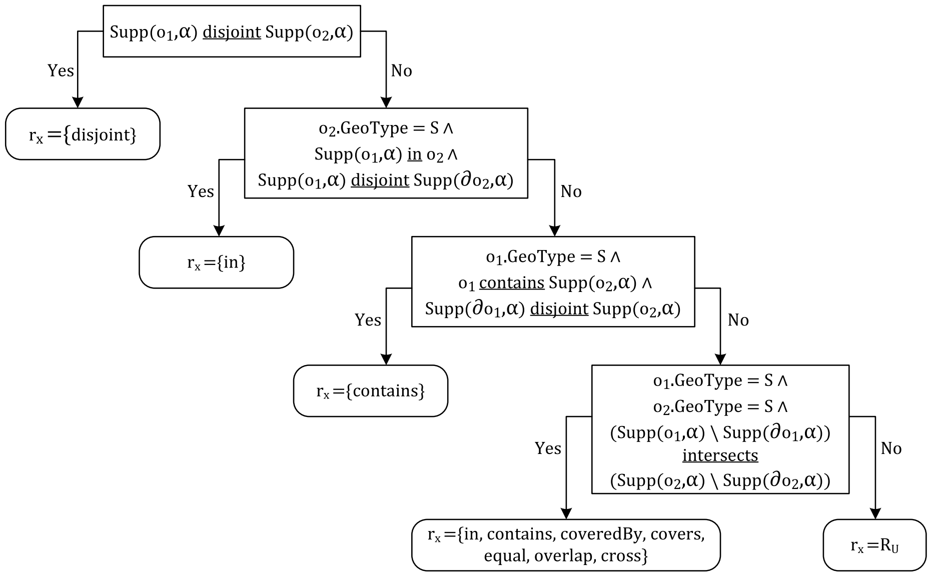


Figure 2: Algorithm for deriving the topological relation between two objects starting from the relation between their supports.

Method 2 (Relation integration)**.** Given two distinct MACS databases DB^1_{macs}, DB^2_{macs}, the integration of the sets of topological relations (or logic observations) that are known in each of them is represented by the function $logicRelIntegration(REL_1, OBJ_1, REL_2, OBJ_2)$. In order to obtain this result we first compute the complete set of relations known by DB^1_{macs} (DB^2_{macs}), denoted as $R_1 = ext(REL_1, OBJ_1)$ ($R_2 = ext(REL_2, OBJ_2)$), and, starting from them, we compute R_3 as follows (referring to Table 3 for the cases definition and to Table 1 for relation symbols):

- in cases $A.0$, $A.1$, $A.2$, $A.3$ and $B.1$, $B.2$ no objects are shared by the integrating databases DB^1_{macs} and DB^2_{macs}:

$$R_3 = R_1 \cup R_2 \;\cup\; topFromSupp(R_1, OBJ_1, R_2, OBJ_2)$$

  where $topFromSupp(R_1, OBJ_1, R_2, OBJ_2)$ has been introduced in Obs. 4.

- in cases $A.4$, $A.5$ and $B.3$, $B.4$, $B.5$ there are some common objects between the integrating databases, thus the function works differently:

$$\begin{aligned} R_3 =& (R_1 \setminus_{ID} R_2) \cup (R_2 \setminus_{ID} R_1) \cup \\ & topFromSupp(R_1, OBJ_1 \setminus_{ID} OBJ_2, R_2, OBJ_2 \setminus_{ID} OBJ_1) \cup \\ & mergeTopRel(R_1, OBJ_1 \cap_{ID} OBJ_2, R_2, OBJ_2 \cap_{ID} OBJ_1) \end{aligned}$$

  where $(R_i \setminus_{ID} R_j) = \{\langle a, b, r_x\rangle \mid \langle a, b, r_x\rangle \in R_i \wedge \langle a, b, r_y\rangle \notin R_j\}$ and $(OBJ_i \setminus_{ID} OBJ_j) = \{o \mid o \in OBJ_i \wedge o.ID \notin OBJ_j.ID\}$.

Finally, the function $mergeTopRel(R_1, O_1, R_2, O_2)$ is defined as follows:

$$mergeTopRel(R_1, O_1, R_2, O_2) = \{\langle o_1, o_2, r\rangle \mid o_1 \in O_1 \wedge o_2 \in O_2 \wedge \\ \langle o_1, o_2, r_1\rangle \in R_1 \wedge \langle o_1, o_2, r_2\rangle \in R_2 \wedge r = r_1 \cap r_2)\} \quad (16)$$

The result of the function $logicRelIntegration(REL_1, REL_2) = REL_3$ is obtained by considering the entries of R_3 that represents disjunction of relations or empty relations. □

Notice that, the *mergeTopRel* function can produce empty relations (as result of the intersection $r_1 \cap r_2$); these empty relations represent inconsistencies between the integrating databases and have to be solved by human intervention.

The human intervention is necessary whenever the logic observations contained in the source databases are contrasting. However, if the cost of human intervention is too high or the user is not able to determine the right relation for the final database, then some automatic procedures can be implemented in order to convert the inconsistency into a loss of accuracy. For this purpose we consider the proximity relationship among topological relations first introduced in [EM95] for the definition of conceptual neighborhoods starting from the the 9-intersection matrices. This definition has been extended in [BCP05] in order to be applied to relations defined by means of sets of 9-intersection matrices, as those defined in Table 1. In particular, the distance between two relations is computed considering the minimum distance between the corresponding 9-intersection matrices. In this report we adopt the same approach for defining, given a topological relation rel_1 between specific object types (for example, between surfaces), the set of relations that are near to it. We say that a topological relation rel_1 is *near* another relation rel_2, if rel_2 is characterized by a matrix with the minimum distance (variation) with respect to the matrix characterizing rel_1. The following definition formally specifies the proximity between topological relations. Fig. 3 illustrates proximity between topological relations calculated on the basis of the type of the involved objects. An arc is depicted between two topological relations if they are near and the label on each arc denotes the distance between them. Let us notice that when a topological relation have several matrices associated to it, each of these can have different distances with respect to the matrix of another relation but for simplicity only the minimum distance is reported in the diagram.

Definition 8 (Topological relation proximity)**.** Given two topological relations rel_1 and rel_2 both defined between objects of type t_1 and t_2; and represented by the set of 9-intersection matrices M_1 and M_2, respectively. We say that rel_1 is *near* rel_2 if

$$distance(rel_1, rel_2) = min(distance(rel_1, r) \mid r \neq rel_1 \wedge r \in R_{topo})$$

where $distance()$ calculates the distance between two topological relations as the minimum number of discordant elements between two matrices $m_1 \in M_1$ and $m_2 \in M_2$.

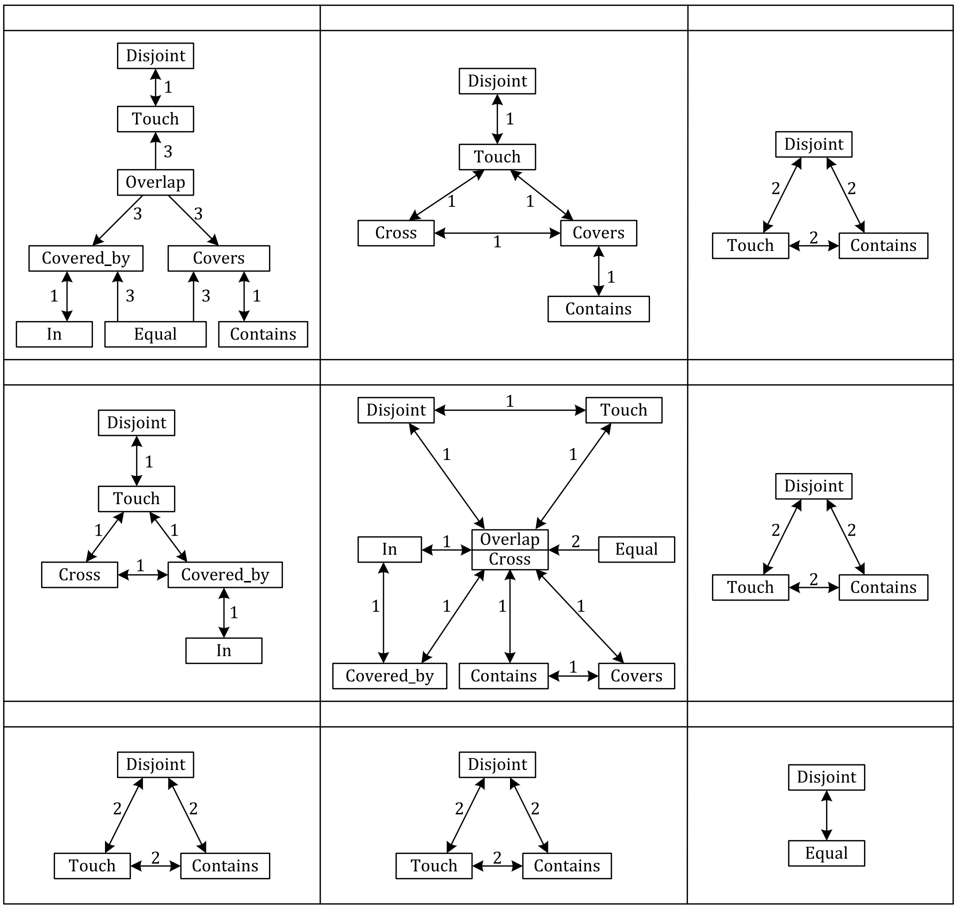


Figure 3: Proximity between topological relations classified on the basis of the type of the involved objects. Let us notice that for not cluttering the diagram, the relations *Overlap* and *Cross* between two curves have been collapsed into a unique box because they have the same distance from the other relations. The distance between them is 1 if we consider the dimension of the intersection between their interior.

If this kind of approach can be admissible for the user, we can assume that when the topological relations in the source databases are not compatible but are near with respect to Def. 8, then the resulting relation becomes the union of the original ones. Formally we can obtain this result by substituting the intersection $r_1 \cap r_2$ in Eq. 16 with $near(r_1, o_1.Geo.type, o_2.Geo.type) \cap near(r_2, o_1.Geo.type, o_2.Geo.Type)$, where $near(r, t_1, t_2)$ computes the set of relations that are near to r when objects of types t_1 and t_2 are considered.

4.3 Integrating Metric and Logic Observations

The complete integration of two MACS databases requires to combine metric and logic observations together. In particular, in Sec. 3.2 we have introduced

Operation	**Syntax**	**Precondition**	**Semantics**
positions snapping	$a \rightarrow\leftarrow b$	$\exists P : P \in a.Geo.pos$, $Supp_P(\alpha) \cap Supp(b, \alpha) \neq \emptyset \vee$ $\exists Q : Q \in b.Geo.pos$, $Supp(a, \alpha) \cap Supp_Q(\alpha) \neq \emptyset$	identify the pair of positions (P', Q') to snap (they can be existing positions or positions generated by projection), substitute Q' with P' in b and consider Q' as a new observation of the position of P' to be integrated.
one position snapping	$a \rightarrow\leftarrow_1 b$	as for $a \rightarrow\leftarrow b$	as for $a \rightarrow\leftarrow b$, but only one substitution is admitted.
two positions snapping	$a \rightarrow\leftarrow_2 b$	as for $a \rightarrow\leftarrow b$ and: $\exists P : P \in a.Geo.pos$, $Supp_P(\alpha) \cap Supp(b, \alpha) = \emptyset \vee$ $\exists Q : Q \in b.Geo.pos$, $Supp(a, \alpha) \cap Supp_Q(\alpha) = \emptyset$	as for $a \rightarrow\leftarrow b$, but exactly two subsequent positions must be snapped.
right positions snapping	$a \rightarrow\leftleftarrows b$	$\forall Q \in b.Geo.pos :$ $Supp(a, \alpha) \cap Supp_Q(\alpha) \neq \emptyset \wedge$ $\forall P \in match(a, b) :$ $Supp(b, \alpha) \cap Supp_P(\alpha) \neq \emptyset$	for all $Q_i \in b.Geo.pos$ identify the pair of positions (P_i, Q_i) to snap, substitute Q_i with corresponding $P_i \in a.Geo.pos$ and consider positions Q_i new observations of the positions P_i to be integrated.
all positions snapping	$a \rightrightarrows\leftleftarrows b$	$\forall P \in a.Geo.pos :$ $Supp_P(\alpha) \cap Supp(b, \alpha) \neq \emptyset \wedge$ $\forall Q \in b.Geo.pos :$ $Supp(a, \alpha) \cap Supp_Q(\alpha) \neq \emptyset$	identify the pairs of positions (P_i, Q_i) to snap (no positions of a or b have to remain dangling), substitute each Q_i with corresponding P_i and consider Q_i new observations of the positions P_i to be integrated.

Table 4: Positions snapping operations. $match(a, b)$ returns all the positions of a that have a matching with a position of b or that are between two matching positions.

the coherence constraint between soft topological relations, which are those derived from object representatives, and hard topological relations, which are those explicitly stored. Moreover, in the same section we have established that for reducing the quantity of stored information, when the topological relation is known, it can be derived directly from the geometries of the objects representatives without additional information.

However, in general, after the integration operations presented in the previous sections a check phase is necessary in order to verify that the coherence constraint is satisfied in the resulting MACS database DB^3_{macs}. This means that for each pair of objects of OBJ_3 the soft topological relation between them has to be compute, denoted as r_{soft}, and compare it with the relation eventually stored in REL_3, denoted as R. If $r_{soft} \in R$ then the coherence constraint is satisfied, otherwise it is necessary to modify the positions defining the objects geometries, in order to obtain a new situation where the r_{soft} changes and becomes one of the relations in R. Indeed, we always suppose that the logic observations have priority with respect to metric observations.

The remainder of this section analyses how metric observations compliant with a topological relation rel_1 have to be transformed in order to become compliant with another desired topological relation rel_2. In doing so, let us notice that some transitions from one topological relation to another, like the transition $disjoint \rightarrow touch$, require that two distinct positions of the two objects in relation becomes the same position. We denote this case with the term *positions snapping* ($\rightarrow\leftarrow$). For other transitions the inverse operation is required, i.e. a shared position has to be transformed into two distinct ones. We denote this operation as *positions decoupling* ($\leftarrow\rightarrow$). Finally, in some cases the switch of location for a position with respect to a curve or surface is necessary. This operation is denoted as *positions switching* ($\rightleftarrows$). In Tables 4, 5 and 6 the operations semantics and the necessary preconditions for their application are summarized.

Method 3 (Alignment of positions with respect to logic observations)**.** Given an integrated MACS databases DB^3_{macs} and the initial databases DB^1_{macs} and DB^2_{macs} the alignment of positions with respect to logic observations is an iterate process that is executed until the following condition holds:

$$\begin{aligned} \{(o_1, o_2) \quad | \quad & (o_1, o_2) \in OBJ_3 \times OBJ_3 \wedge o_1 \; r_{soft} \; o_2 \; \wedge \\ & \langle o_1, o_2, R \rangle \in REL_3 \wedge r_{soft} \notin R\} = \emptyset \end{aligned} \tag{17}$$

The core algorithm, that is reiterated, is composed of the following tasks:

- for each consistency violation between a pair of objects (o_1, o_2), the necessary relation transition $r_A \rightarrow r_B$ is identified;
- for each relation transition its applicability is evaluated; in particular, some transition are not admitted a priori, some others require operations that, in specific cases, could not be applied (Tables 4-6 in this section and Tables 7-15 in Appendix);
- for each relation transition that is not applicable, since its preconditions are not satisfied, the user intervention is requested;
- for each relation transition $r_A \rightarrow r_B$ that is applicable and such that $o_1 \; r_{soft} \; o_2$ in DB^i_{macs} ($i \in \{1, 2\}$) and $r_{soft} = r_B$, we augment the accuracy of the relative distance among all the position pairs $(P_i, Q_i) \in o_1.Geo.pos \times o_2.Geo.pos$ having intersecting supports by setting the covariance equals to the value $(\sigma^2_P + \sigma^2_Q)/2$ (see Eq.6) in

the corresponding matrix C_{DB_i}. Then we repeat the computation of $DB_3 = metricPosIntegration_{kalman}(DB_1, DB_2, C_{DB_1}, C_{DB_2})$.

- for each relation transition $r_A \rightarrow r_B$ that is applicable but does not satisfy the previous condition, it is necessary to modify some pairs of positions $(P_j, Q_j) \in o_1.Geo.pos \times o_2.Geo.pos$ having intersecting supports, by applying the operations requested by the transition, as shown in Tables 7-15 in Appendix.. This leads to the definition of a new $DB_3^{'}$ and to a new variance-covariance matrix $C_{DB_3^{'}}$ that needs to be integrated with DB_3 in order to obtain the final database: $DB_{final} = metricPos\ Integration_{kalman}(DB_3, DB_3^{'}, C_{DB_3}, C_{DB_3^{'}})$. $\square$

Notice that, in Tables 7-15 some allowed transitions involve pairs of relations that are not near. These cases are considered since the transition can be obtained with a local geometry modification, i.e. by applying a minimal change on objects positions.

The main idea underlying phases 4 and 5 is: the positions of the objects involved into a particular topological relation shall become a rigid body that can move in space but in a uniform manner: they have to maintain their relative reciprocal positions in order to keep the effect of the previous transformations. This is the aim of the covariance correction the is proposed in phase 4 and in the operations eventually applied in phase 5.

Notice that our approach is different with respect to the one presented in [Hop08] [HK08]; first of all, we consider the integration of both metric and topological information, while they suppose to have only one set of topological relation that has to be valid on the integrated geometry. We cannot use sets of equations representing the topological relations that are valid in the two source datasets, because if they contains discordant information the method cannot find a solution that satisfy all the equations. Moreover, our method consider not only single relation, but also sets (disjunctions) of topological relations between objects, so the number of necessary equations, that have to be added into the system in the approach of [Hop08][HK08], can increase considerably making the integration impracticable. Finally, thanks to the role covered by the accuracy of the relative distances, most of the topological relations that are valid before the update, remain satisfied also in the integrated database: typically in practice very few relations are violated after the integration process.

In oder to prove that the proposed operations (shown in Tables 7-15) are a sufficient condition for obtaining the needed relation transitions, we show below the proof of this property for the transitions that start from a disjoint relation. In a similar way the same property can be proved for other transitions.

Theorem 1 (Operations for disjoint transitions)**.** *Let us consider the first part of Table 7, showing the allowed transitions starting from the disjoint*

Operation	Syntax	Precondition	Semantics
in positions decoupling	$a \overset{in}{\longleftrightarrow} b$	$\exists P:$ $P \in a.Geo.pos$, $P \in b.Geo.pos$	substitute P with two new positions $Q_1 \in a.Geo.pos$ and $Q_2 \in b.Geo.pos$, where the distance between Q_1 and Q_2 is the minimum representable distance ϵ such that Q_2 *in* a and Q_1 *in* b. Finally, the accuracy of the relative distance between Q_1 and Q_2 is maximized.
in left positions decoupling	$a \overset{in_L}{\longleftrightarrow} b$	as for $a \overset{in}{\longleftrightarrow} b$	as for $a \overset{in}{\longleftrightarrow} b$, but requiring that Q_1 *in* b and Q_2 *disjoint* a.
out positions decoupling	$a \overset{out}{\longleftrightarrow} b$	as for $a \overset{in}{\longleftrightarrow} b$	as for $a \overset{in}{\longleftrightarrow} b$, but requiring that Q_1 *disjoint* b and Q_2 *disjoint* a
cross positions decoupling	$a \overset{cr}{\longleftrightarrow} b$	as for $a \overset{in}{\longleftrightarrow} b$	as for $a \overset{in}{\longleftrightarrow} b$, but requiring that a *cross* b after decoupling.
all out (in left) positions decoupling	$a \overset{*}{\leftleftarrows\!\rightrightarrows} b$	as for $a \overset{in}{\longleftrightarrow} b$	as for $a \overset{in_L}{\longleftrightarrow} b$ (or $a \overset{out}{\longleftrightarrow} b$), but requiring that all sharing positions are decoupled and that, after the operation, the relation a *in* b (or a *disjoint* b) is satisfied.

Table 5: Positions decoupling operations. (The distance ϵ could be a parameter set by the user, however it has always to be significantly lower w.r.t. the average error of absolute coordinates).

Operation	Syntax	Precondition	Semantics
in positions switching	$a \overset{in}{\rightleftarrows} b$	as for $a \rightarrow\leftarrow b$	it is the combination of $a \rightarrow\leftarrow b$ followed by $a \overset{in}{\longleftrightarrow} b$.
out positions switching	$a \overset{out}{\rightleftarrows} b$	as for $a \rightarrow\leftarrow b$	it is the combination of $a \rightarrow\leftarrow b$ followed by $a \overset{out}{\longleftrightarrow} b$.
cross positions switching	$a \overset{cr}{\rightleftarrows} b$	as for $a \rightarrow\leftarrow b$	it is the combination of $a \rightarrow\leftarrow b$ followed by $a \overset{cr}{\longleftrightarrow} b$.
all in positions switching	$a \overset{in}{\rightleftarrows}_{all} b$	as for $a \rightarrow\leftleftarrows b$	it is the combination of $a \rightarrow\leftleftarrows b$ followed by $a \overset{in}{\longleftarrow\!\rightrightarrows} b$
all out positions switching	$a \overset{out}{\rightleftarrows}_{all} b$	as for $a \rightarrow\leftleftarrows b$	it is the combination of $a \rightarrow\leftleftarrows b$ followed by $a \overset{out}{\longleftarrow\!\rightrightarrows} b$

Table 6: Positions switching operations.

relation. Each column, representing a given target relation $rel_{*,*}$, reports for each type pair t_1, t_2 the operations that represent a sufficient condition to obtain, starting from two disjoint objects of types t_1, t_2, the target relation.

Proof. We present the proof for the first column of the table, the proof for the other columns follows a similar reasoning:

Transition (a *disjoint* b) $\rightarrow$ (a *touch* b), for type pairs (S, S), (S, C), (C, C) and (C, S): according to Table 1 the pattern for disjoint in these cases is $FFT - \mathbf{FF}T - TTT$. If we apply the required operation $a \rightarrow\leftarrow b$, when objects types are (S, S), then the geometries of a and b is locally modified, so that after the modification a and b share a position. As a consequence, the intersection $\partial a \cap \partial b$ becomes not empty and the pattern becomes: $FFT - F\mathbf{T}T - TTT$, which is the pattern of the touch relation for types (S, S). When objects types are (S, C), either $\partial a \cap \partial b$ becomes not empty or $\partial a \cap b^\circ$ does, thus the pattern becomes $FFT - F\mathbf{T}T - TTT$ or $FFT - \mathbf{T}FT - TTT$, which again are patterns of touch. A dual reasoning can be applied when objects types are (C, S). When objects types are (C, C), the required operation is $\partial a \rightarrow\leftarrow \partial b$, or $\partial a \rightarrow\leftarrow_1 b$, or $a \rightarrow\leftarrow_1 \partial b$. As a consequence, either $\partial a \cap \partial b$ becomes not empty or $\partial a \cap b^\circ$ ($a^\circ \cap \partial b$) does, thus the pattern becomes $FFT - F\mathbf{T}T - TTT$ or $FFT - \mathbf{T}FT - TTT$ ($F\mathbf{T}T - FFT - TTT$), which again are patterns of touch.

Transition (a *disjoint* b) $\rightarrow$ (a *touch* b), for type pairs (S, P) and (C, P): according to Table 1 the pattern for disjoint in these cases is $FFT - \mathbf{F}FT - TFT$. If we apply the operation $a \rightarrow\leftarrow b$ ($\partial a \rightarrow\leftarrow b$), then $\partial a \cap b$ becomes not empty and $a^- \cap b$ becomes empty, thus the pattern becomes $FFT - \mathbf{T}FT - \mathbf{F}FT$, which is the pattern of touch.

Transition (a *disjoint* b) $\rightarrow$ (a *touch* b), for type pairs (P, S) and (P, C): the reasoning in this case is similar to the previous one. □

5 Properties of the Integration Process

This section presents some properties of the integration method proposed in Sec. 4. First of all, we describe the trend of the coefficients of the Kalman matrix in relation to the different accuracies of the two source databases. Then we discuss the central role covered by the accuracy of each measure during the integration. We show that the shift of a position from its original location depends not only upon the value of the integrated measures, but also on the accuracy of these measures and its correlation with near positions. Finally, we state that the accuracy of the integrated metric and the certainty of the logic observations are always increased after the integration process or at least coincide with the accuracy and certainty of the most accurate source database.

Property 1. Given two MACS database DB^1_{macs} and DB^2_{macs} that have to be integrated, the coefficients of the Kalman matrix associate to each absolute or relative measure in DB^2_{macs} a value proportional to its accuracy and normalised with respect to the overall accuracy of the two source databases. In particular, the coefficients of the Kalman matrix related to x_P assume a value as follows:

- The coefficient k_{x_P,x_P} related to the variance of P has a value between 0 and 1.

$$k_{x_P,x_P} = \begin{cases} a \in [0, 0.5) & \text{if } DB^2_{macs} <_{acc} DB^1_{macs} \\ a = 0.5 & \text{if } DB^2_{macs} =_{acc} DB^1_{macs} \\ a \in (0.5, 1] & \text{if } DB^2_{macs} >_{acc} DB^1_{macs} \end{cases} \tag{18}$$

- The coefficients k_{x_P,x_Q} for $Q \neq P$ related to the covariance between the x coordinate of P and the x coordinate of another point Q have a value between -1 and 1.

$$k_{x_P,x_Q} = \begin{cases} b \in [-1, 0) & \text{if } DB^2_{macs} <_{acc} DB^1_{macs} \\ b = 0 & \text{if } DB^2_{macs} =_{acc} DB^1_{macs} \\ b \in (0, 1] & \text{if } DB^2_{macs} >_{acc} DB^1_{macs} \end{cases} \tag{19}$$

- The coefficients k_{x_P,y_Q} for $Q \neq P$ corresponding to the covariance between the x coordinate of P and the y coordinate of another point Q are zero.

From these characteristics of the coefficients of the Kalman matrix, we can state the first property of the integration process.

Property 2. Given two MACS databases DB^1_{macs} and DB^2_{macs} that have to be integrated, the shift of a position P in DB^1_{macs} increases if the accuracy of P in DB^2_{macs} is greater than the accuracy of P in DB^1_{macs}.

Proof. Given the vectors V_{DB_1} and V_{DB_2} built as explained in Obs. 3, the vector of position indexes V_{DB_3} for the integrated database DB^3_{macs} is obtained using the Eq. 14 as follows:

$$V_{DB_3} = V_{DB_1} + K \cdot (V_{DB_2} - V_{DB_1})$$

Let us suppose for simplicity that inside the two source databases there are only two positions $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$. The shift of the integrated x coordinate of position P, denoted as x^3_P, from its original value in DB^1_{macs} becomes:

$$x^3_P - x^1_P = k_{x_P,x_P}(x^2_P - x^1_P) + k_{x_P,x_Q}(x^2_Q - x^1_Q)$$

where $k_{i,j}$ is the coefficient of the Kalman matrix in row i and column j. The terms related to the y coordinates can be eliminated since the x and y

coordinates are considered mutually independent and so the correspondent elements in the Kalman matrix is zero. Independently from the measurements contained in DB^2_{macs} (x^2_P and x^2_Q), the shift of x^3_P from its original value x^1_P directly depends upon the coefficient k_{x_P,x_P} and k_{x_P,x_Q} of the Kalman matrix. Property 1 states that more the accuracy of the position P in DB^2_{macs} increases with respect to the accuracy of the same position in DB^1_{macs}, more the value of the coefficients k_{x_P,x_P} and k_{x_P,x_Q} tend to one, determining a greater shift of x^3_p that can eventually become equal to x^2_P. Let us notice that the shift of P is due not only to a direct update of its measure in DB_2, but also to the propagation of the update of other positions, in a measure that directly depends upon the accuracy of the relative distance between them. □

Example 3. Let us suppose that DB^1_{macs} contains two positions $P = (100, 100)$ and $Q = (123, 123)$ that have both an absolute accuracy e of 0.8 meters (with $F_R(e) = 95\%$), while their relative distance has an accuracy of 0.6 meters (with $F_R(e_R)= 95\%$). Moreover, DB^2_{macs} contains another measure for $P = (103, 103)$ that has to be integrated with the one contained in DB^1_{macs}. We perform the integration between the measures of the two source databases varying the error $e(P^2)$ of P in DB^2_{macs} and we analyze the different shift of P and Q in DB^3_{macs} from their original positions in DB^1_{macs}. The results of this test are reported in the graph of Fig. 4. The graph clearly illustrates that grater is the accuracy of x_P in DB_2 (smaller is its circular error), grater is the shift of both points after the integration process. Moreover, even if the trend for the two points is similar, the shift of P is greater because it is directly involved in the integration process, while the shift of Q is due only to the propagation of the P integration.

Property 3. Given the MACS database DB^3_{macs} obtained by integrating two source MACS databases DB^1_{macs} and DB^2_{macs}, the accuracy of each integrated measure in DB^3_{macs} is not smaller than the accuracy of the corresponding measure in the two source databases. In particular, if the accuracy of a measure in one database is very high, then the corresponding measure in the other database does not influence the integration process and the resulting accuracy corresponds to the greatest one.

Proof. The metric accuracy of a position P is defined in Eq. 8 and it inversely depends on the position variance. The variance for the integrated position P in DB^3_{macs} is computed using the Eq. 15 as:

$$C_{DB_3} = (I - K) \cdot C_{DB_1} \cdot (I - K)^T + K \cdot C_{DB_2} \cdot K^T$$

Let us suppose that DB^2_{macs} contains a very accurate measure for P, then from Property 1 it results that the coefficient k_{x_P,x_P} (or equivalently k_{y_P,y_P}) of the Kalman matrix has a value near one. From this, it follows that the

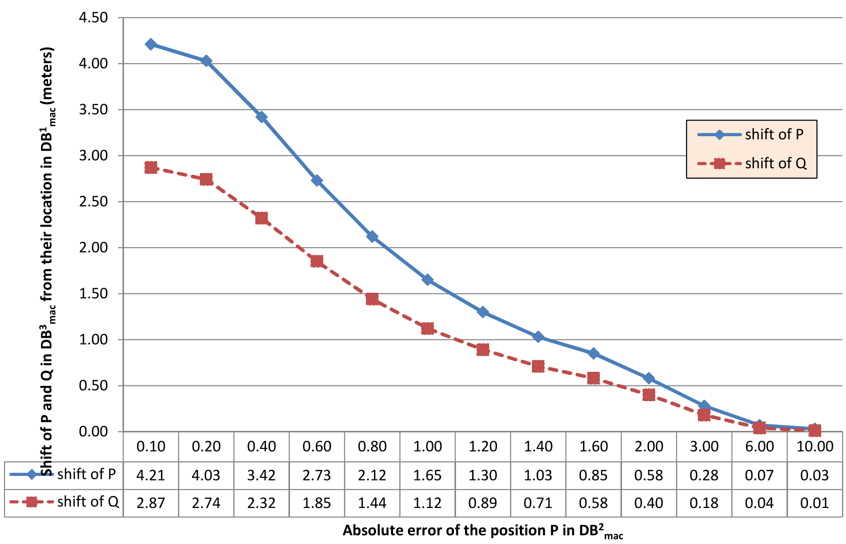
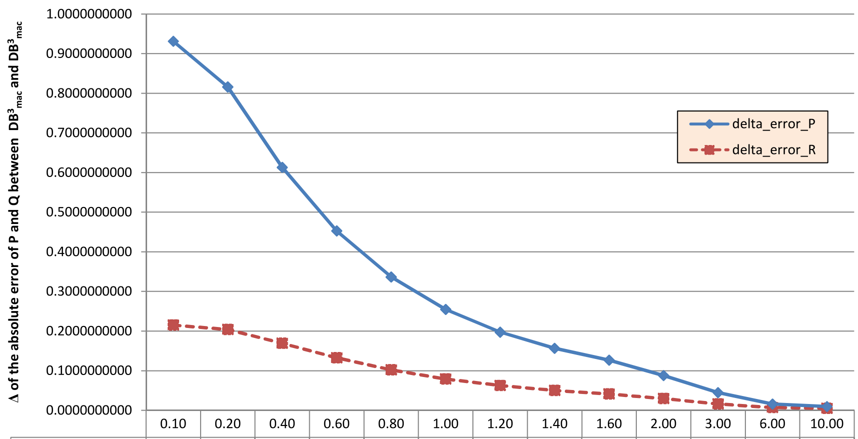


Figure 4: Shift of the positions P and Q with respect to their original measures in DB^1_{macs}, considering different absolute error $e(P^2)$ for P in DB^2_{macs}.

resulting variance value in C_{DB_3} is very closed to (at most coincides with) the element contained in C_{DB_2}, namely to the most accurate one. Conversely, if DB^2_{macs} contains a very accurate measure for P, then the coefficient k_{x_P,x_P} of the Kalman matrix has a value near zero and the variance value in C_{DB_3} for P is very closed to the one in C_{DB_1}. In the other cases, if the two source databases contain both relative accurate measures for P, the diagonal position $k_{P,P}$ of the Kalman matrix contains a positive but smaller than one element. This element multiplied with the elements of the original matrices produces values that are smaller than the original ones; moreover, their sum is smaller than each original value as the coefficient of K are normalised with respect to the overall accuracy of the two databases (the sum of the two original variances). Finally, as the variance of each measure decreases at each iteration, the quality of the integrated position always increase. $\square$

Example 4. Let us consider again the two MACS databases in Example 3 and perform the integration between them taking into account the new value of absolute error calculated after the integration process. The error values for the integrated measures is reported in the graph of Fig. 5, considering different values for the absolute error $e(P^2)$ of the position P in DB^2_{macs}. We can notice that as P is more accurate, then the error of the integrated measures decreases. Moreover, if the error $e(P^2)$ is equal to the error $e(P^1)$ of P in DB_1 (0.80 meters), then the integrated measure has an error that is smaller then the original ones: the integration of two measures with the same accuracy produces a new measure that is more accurate than the two source ones. Finally, if the measure of P is very inaccurate, then it has not effect during the integration process also as regards to the error of the integrated measure, indeed as $e(P^2)$ increases the resulting error for the integrated measure settles to a value near the original error in DB^1_{macs} (0.80 meters).



	0.10	0.20	0.40	0.60	0.80	1.00	1.20	1.40	1.60	2.00	3.00	6.00	10.00
delta_error_P	0.7164	0.6121	0.4437	0.3202	0.2343	0.1754	0.1346	0.1059	0.0852	0.0582	0.0284	0.0087	0.0043
delta_error_R	0.2150	0.2039	0.1693	0.1325	0.1021	0.0792	0.0626	0.0504	0.0414	0.0296	0.0161	0.0071	0.0051

Absolute error of the position P in DB^2_{mac}

Figure 5: Variation of the absolute error for the integrated positions P and Q in DB^3_{macs} with respect to the value in DB^1_{macs}, considering different absolute error for P in DB_2.

Property 4. The average global accuracy concerning the metric observations of an integrated MACS database DB_{macs} is always grater than the average global accuracy concerning the metric observations of both the source databases.

Proof. The proof directly derives from the definition of average global accuracy of metric observations given in Eq. 9 and from the previous property. In particular, the global metric accuracy of a MACS database is equal to the average of the accuracies of each point and as these accuracies increase at every integration, then also their average increase at every integration. □

Property 5. Given a MACS database DB^3_{macs} obtained by integrating two source MACS databases DB^1_{macs} and DB^2_{macs}, the certainty of each logic observations does not decrease during the integration process, it can only remains unchanged or increases.

Proof. The certainty of each logic observations is defined in Eq. 10. Discarding the optimization mentioned at the end of Sec. 4.2, given two disjunction of topological relations R^1_n and R^2_n, their integration always produces a set of relations R_n^3 whose cardinality is smaller than the cardinality of both the original ones, or equal to the smallest ones ($|R_n^3| \leq min(|R_n^1|, |R_n^2|)$. Therefore, putting this new cardinality into the certainty formula, we obtain a certainty index that is equal to the grater one or is grater than both the original ones. □

Property 6. The average global certainty concerning logic observations of an integrated MACS database is always grater than or equal to the maximum average global certainty of the source databases.

Proof. The average global certainty concerning logic observations is defined in Eq. 11. First of all, we can observe that the number of objects contained into the integrated database is always grater or equal to the number of objects contained in the two source databases. Moreover, some of the unknown relations can become known during the integration process but none of the known relation can become unknown. Finally, as proved above the certainty index of the unknown relations augments at each integration. □

From the presented properties we can conlude that the proposed integration process does not decrease (and usually increases) the overall knowledge of a certain geographical area represented in a MACS database with respect to both metric and logic observations.

6 Conclusions

The integration of spatial data is an important activity, especially in an open and distributed environment, such a Spatial Data Infrastructure (SDI). Spatial data is inherently characterized by some accuracy parameters that have to be considered during an integration process. Unfortunately, it is not a common practice to attach accuracy information to the spatial data stored inside a spatial database.

In this paper we proposed a model for representing a multi-accuracy spatial database, called MACS, and we discuss how accuracy values can be derived from the commonly available information stored inside a spatial database. Then we proposed a methodology for integrating two MACS databases containing metric and logic observations and we discussed how these two kind of information can be combined and kept consistent in the resulting database. The proposed methodology allows not only to integrate metric observations and maintaining them consistent with the desired topological relations, but also provides an accuracy estimate for the resulting database. Finally, some properties of the proposed integration procedure are presented, they principally illustrate how considering the accuracy of measures can affect the resulting integrated dataset and its resulting accuracy.

A Appendix

This section contains the tables explaining all the possible transitions between topological relations. In each cell is reported in round brackets the distance between the two considered topological relations ("*req. d*" means that the transition is allowed only when the distance between the matrix of the current scene and the requested relation *rel* is *d*) and below the operations that have to be applied in order to obtain the requested relation. The symbol *ND* indicates that the target relation *rel* is not defined for the considered geometric types, while *NA* indicates that *rel* cannot be obtained without a human intervention.

$a\ d_{*,*}\ b \rightarrow a\ rel\ b$								
$d_{*,*}$	*rel*							
	$t_{*,*}$	$i_{*,*}$	$c_{*,*}$	$e_{*,*}$	$r_{*,*}$	$o_{*,*}$	$b_{*,*}$	$v_{*,*}$
$d_{S,S}$	(1) $a \rightarrow\leftarrow b$	NA	NA	NA	ND	(4) $a \overset{in}{\rightleftarrows} b$	NA	NA
$d_{S,C}$	(1) $a \rightarrow\leftarrow b$	ND	NA	ND	(2) $a \overset{in}{\rightleftarrows} b$	ND	ND	NA
$d_{S,P}$	(2) $a \rightarrow\leftarrow b$	ND	(2) $a \overset{in}{\rightleftarrows}_{all} b$	ND	ND	ND	ND	ND
$d_{C,S}$	(1) $a \rightarrow\leftarrow b$	NA	ND	ND	(2) $a \overset{in}{\rightleftarrows} b$	ND	NA	ND
$d_{C,C}$	(1) $\partial a \rightarrow\leftarrow_1 b$ or $a \rightarrow\leftarrow_1 \partial b$ or $\partial a \rightarrow\leftarrow \partial b$	NA	NA	NA	(1) $a \overset{cr}{\rightleftarrows} b$	(1) $a \rightarrow\leftarrow_2 b$	NA	NA
$d_{C,P}$	(2) $\partial a \rightarrow\leftarrow b$	ND	(2) $a^\circ \rightarrow\leftleftarrows b$	ND	ND	ND	ND	ND
$d_{P,S}$	(2) $a \rightarrow\leftarrow b$	(2) $a \overset{in}{\rightleftarrows}_{all} b$	ND	ND	ND	ND	ND	ND
$d_{P,C}$	(2) $\partial a \rightarrow\leftarrow b$	(2) $a^\circ \rightarrow\leftleftarrows b$	ND	ND	ND	ND	ND	ND
$d_{P,P}$	ND	ND	ND	(3) $a \rightrightarrows\leftleftarrows b$	ND	ND	ND	ND

Table 7: Transitions between topological relations: case $disjoint \rightarrow rel$.

References

[Alt93] R. B. Altman. A probabilistic algorithm for calculating structure: Borrowing from simulated annealing. In *Ninth Annual Conference on Uncertainty in Artificial Intelligence*, Washington, D. C., 1993. Morgan Kaufman.

[BCP05] Alberto Belussi, Barbara Catania, and Paola Podestà. Towards topological consistency and similarity of multiresolution geo-

$a\ e_{*,*}\ b \rightarrow a\ rel\ b$								
$e_{*,*}$	*rel*							
	$d_{*,*}$	$t_{*,*}$	$i_{*,*}$	$c_{*,*}$	$r_{*,*}$	$o_{*,*}$	$b_{*,*}$	$v_{*,*}$
$e_{S,S}$	NA	NA	NA	NA	ND	NA	(3) $a \xleftrightarrow{in} b$	(3) $a \xleftrightarrow{out} b$
$e_{C,C}$	NA	NA	NA	NA	NA	(2) $a \xleftrightarrow{out} b$	(3) $\partial a \xleftrightarrow{out} b$	(3) $\partial b \xleftrightarrow{out} a$
$e_{P,P}$	(3) $a \xleftrightarrow{out} b$	ND	ND	ND	ND	ND	ND	ND

Table 8: Transition between topological relations: case *equal* $\rightarrow$ *rel*.

$a\ t_{*,*}\ b \rightarrow a\ rel\ b$								
$t_{*,*}$	*rel*							
	$d_{*,*}$	$i_{*,*}$	$c_{*,*}$	$e_{*,*}$	$r_{*,*}$	$o_{*,*}$	$b_{*,*}$	$v_{*,*}$
$t_{S,S}$	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$	NA	NA	NA	ND	(3) $a \xleftrightarrow{in_L} b$	NA	NA
$t_{S,C}$	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$	ND	NA	ND	(req. 1) $a \xleftrightarrow{cr} b$	ND	ND	(req. 1) $a \xleftrightarrow{in_L} b$
$t_{S,P}$	(2) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$	ND	(2) $a \xleftrightarrow{in_L} b$	ND	ND	ND	ND	
$t_{C,S}$	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$	NA	ND	ND	(req. 1) $a \xleftrightarrow{cr} b$	ND	(req. 1) $b \xleftrightarrow{in_L} a$	ND
$t_{C,C}$	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$	NA	NA	NA	(1) $a \xleftrightarrow{cr} b$	(1) $a \rightarrow\leftarrow_2 b$	NA	NA
$t_{C,P}$	(2) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$	ND	(2) $a \xleftrightarrow{in_L} b$	ND	ND	ND	ND	ND

Table 9: Transition between topological relations: case *touch* $\rightarrow$ *rel*.

graphical maps. In *GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 220–229, New York, NY, USA, 2005. ACM.

[BLR+04] B. Bhanu, R. Li, C. Ravishankar, M. Kurth, and J. Ni. Indexing structure for handling uncertain spatial data. In *Sixth International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, 2004.

[CCHF+98] Maria A. Cobb, Miyi J. Chung, III Harold Foley, Frederick E. Petry, Kevin B. Shaw, and H. Vincent Miller. A rule-based approach for the conflation of attributed vector data. *Geoinformatica*, 2(1):7–35, 1998.

[CFvO93] Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *SSD93: Proceedings of the 3rd International Symposium on Advances in Spatial Databases*, pages 277–295. Springer-Verlag, 1993.

$a\ i_{*,*}\ b \rightarrow a\ rel\ b$								
$i_{*,*}$	*rel*							
	$d_{*,*}$	$t_{*,*}$	$c_{*,*}$	$e_{*,*}$	$r_{*,*}$	$o_{*,*}$	$b_{*,*}$	$v_{*,*}$
$i_{S,S}$	NA	NA	NA	NA	ND	(4) $a \overset{out}{\rightleftarrows} b$	(1) $a \rightarrow\leftarrow b$	NA
$i_{C,S}$	NA	NA	ND	ND	(2) $a \overset{out}{\longleftrightarrow} b$	ND	(1) $a \rightarrow\leftarrow b$	ND
$i_{P,S}$	(2) $a \overset{out}{\rightleftarrows} b$	(2) $b \rightarrow\leftleftarrows a$	ND	ND	ND	ND	ND	ND
$i_{C,C}$	NA	NA	NA	NA	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a^\circ \rightarrow\leftarrow_1 b^\circ$	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a \rightarrow\leftarrow_2 b$	(1) $\partial a \rightarrow\leftarrow \partial b$	NA
$i_{P,C}$	(2) $a \overset{out}{\longleftrightarrow} b$	(2) $a \rightarrow\leftarrow \partial b$	ND	ND	ND	ND	ND	ND

Table 10: Transition between topological relations: case *in* $\rightarrow$ *rel*.

$a\ c_{*,*}\ b \rightarrow a\ rel\ b$								
$c_{*,*}$	*rel*							
	$d_{*,*}$	$t_{*,*}$	$i_{*,*}$	$e_{*,*}$	$r_{*,*}$	$o_{*,*}$	$b_{*,*}$	$v_{*,*}$
$c_{S,S}$	NA	NA	NA	NA	ND	(1) $a \overset{out}{\rightleftarrows} b$	NA	(1) $a \rightarrow\leftarrow \partial b$
$c_{S,C}$	NA	NA	NA	ND	(2) $a \overset{out}{\longleftrightarrow} b$	(1)	ND	(1) $a \rightarrow\leftarrow b$
$c_{S,P}$	(2) $a \overset{out}{\rightleftarrows} b$	(2) $a \rightarrow\leftarrow b$	ND	ND	ND	ND	ND	ND
$c_{C,C}$	NA	NA	NA	NA	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a^\circ \rightarrow\leftarrow_1 b^\circ$	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a \rightarrow\leftarrow_2 b$	NA	(1) $\partial a \rightarrow\leftarrow \partial b$
$c_{C,P}$	$a \overset{out}{\longleftrightarrow} b$	$\partial a \rightarrow\leftarrow b$	ND	ND	ND	ND	ND	ND

Table 11: Transition between topological relations: case *contains* $\rightarrow$ *rel*.

[EF91] Max J. Egenhofer and Robert. D. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.

[EF95] Max J. Egenhofer and Robert D. Franzosa. On the Equivalence of Topological Relations. *International Journal of Geographical Information Systems*, 9(2):133–152, 1995.

[EM95] Max J. Egenhofer and David M. Mark. Modelling Conceptual Neighbourhoods of Toplogical Line-Region Relations. *International Journal of Geographical Information Systems*, 9(5):555–565, 1995.

[GGA04a] F. Gielsdorf, L. Gruending, and B. Aschoof. Positional accuracy improvement - a necessary tool for updating and integration of gis data. In *Proceedings of the FIG Working Week 2004*, 2004.

$a\ r_{*,*}\ b \rightarrow a\ rel\ b$								
$r_{*,*}$	*rel*							
	$d_{*,*}$	$t_{*,*}$	$i_{*,*}$	$c_{*,*}$	$e_{*,*}$	$o_{*,*}$	$b_{*,*}$	$v_{*,*}$
$r_{S,C}$	NA	(1) $a \rightarrow\leftleftarrows (a \cap b_P)$	ND	NA	ND	ND	ND	(1) $a \rightarrow\leftleftarrows (b_P \setminus a)$
$r_{C,C}$	NA	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $(\partial a \rightarrow\leftarrow b$ or $a \rightarrow\leftarrow \partial b)$	NA	NA	NA	(1) $a \rightarrow\leftarrow_2 b$	NA	NA

Table 12: Transition between topological relations: case *cross* $\rightarrow$ *rel*. b_P (a_P) is the set of representative points corresponding to the positions used for representing the geometry of b (a).

$a\ o_{*,*}\ b \rightarrow a\ rel\ b$								
$o_{*,*}$	*rel*							
	$d_{*,*}$	$t_{*,*}$	$i_{*,*}$	$c_{*,*}$	$e_{*,*}$	$r_{*,*}$	$b_{*,*}$	$v_{*,*}$
$o_{S,S}$	NA	$a \rightarrow\leftleftarrows (a \cap b_P)$ or $b \rightarrow\leftleftarrows (a_P \cap b)$	NA	NA	NA	NA	$b \rightarrow\leftleftarrows (a_P \setminus b)$ or $a \rightarrow\leftleftarrows (a_P \setminus b)$	$a \rightarrow\leftleftarrows (b_P \setminus a)$ or $b \rightarrow\leftleftarrows (b_P \setminus a)$
$o_{C,C}$	NA	$a \overset{out}{\leftleftarrows\rightrightarrows} b$ $(\partial a \rightarrow\leftarrow b$ or $a \rightarrow\leftarrow \partial b)$	NA	NA	NA	a $\overset{cr}{\longleftrightarrow}$ b	NA	NA

Table 13: Transition between topological relations: case *overlap* $\rightarrow$ *rel*. b_P (a_P) is the set of representative points corresponding to the positions used for representing the geometry of b (a).

[GGA04b] Frank Gielsdorf, Lothar Gruendig, and Bernd Aschoff. Positional accuracy improvement - a necessary tool for updating and integration of gis data. In *Proceedings of the FIG Working Week 2004*, 2004.

[HK08] S. Hope and A. Kealy. Using topological relationships to inform a data integration process. *Transactions in GIS*, 12(2):267–283, 2008.

[HKH06] S. Hope, A. Kealy, and G. Hunter. Improving positional accuracy and preserving topology through spatial data fusion. In *7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences.*, 2006.

[Hop08] S. Hope. *Integration of Vector Datasets.* PhD thesis, Department of Geomatics, University of Melbourne, July 2008.

[LBR+07] R. Li, B. Bhanu, C. Ravishankar, M. Kurth, and J. Ni. Uncertain spatial data handling: Modeling, indexing and query. *Computers and Geosciences*, 33(1):42–61, 2007.

$a\ b_{*,*}\ b \to a\ rel\ b$								
$b_{*,*}$	*rel*							
	$d_{*,*}$	$t_{*,*}$	$i_{*,*}$	$c_{*,*}$	$e_{*,*}$	$r_{*,*}$	$o_{*,*}$	$v_{*,*}$
$b_{S,S}$	NA	NA	(1) $a \overset{in_L}{\leftleftarrows\!\rightrightarrows} b$	NA	(3) $a \rightrightarrows\!\leftleftarrows b$	ND	(1) $a \overset{out}{\longleftrightarrow} b$	NA
$b_{C,S}$	NA	(1) $b \to\!\leftleftarrows a$	(1) $a \overset{in_L}{\leftleftarrows\!\rightrightarrows} b$	ND	ND	$a \overset{out}{\longleftrightarrow} b$	ND	ND
$b_{C,C}$	NA	NA	(1) $\partial a \overset{in_L}{\leftleftarrows\!\rightrightarrows} b$	NA	NA	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$ $a^\circ \to\!\leftarrow_1 b^\circ$	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$ $a \to\!\leftarrow_2 b$	NA

Table 14: Transition between topological relations: case $coveredBy \to rel$.

$a\ v_{*,*}\ b \to a\ rel\ b$								
$v_{*,*}$	*rel*							
	$d_{*,*}$	$t_{*,*}$	$i_{*,*}$	$c_{*,*}$	$e_{*,*}$	$r_{*,*}$	$o_{*,*}$	$b_{*,*}$
$v_{S,S}$	NA	NA	NA	(1) $b \overset{in_L}{\leftleftarrows\!\rightrightarrows} a$	(3) $a \rightrightarrows\!\leftleftarrows b$	ND	(3) $a \overset{out}{\longleftrightarrow} b$	NA
$v_{S,C}$	NA	(1) $a \to\!\leftleftarrows b$	ND	(1) $b \overset{in_L}{\leftleftarrows\!\rightrightarrows} a$	ND	(1) $a \overset{out}{\longleftrightarrow} b$	ND	ND
$v_{C,C}$	NA	NA	NA	(1) $\partial b \overset{in_L}{\leftleftarrows\!\rightrightarrows} a$	NA	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$ $a^\circ \to\!\leftarrow_1 b^\circ$	(1) $a \overset{out}{\leftleftarrows\!\rightrightarrows} b$ $a \to\!\leftarrow_2 b$	NA

Table 15: Transition between topological relations: case $covers \to rel$.

[NRB03] J. Ni, C. V.. Ravishankar, and B. Bhanu. Probabilistic spatial database operations. In *Advances in Spatial and Temporal Databases*, volume LNCS 2750/2003, pages 140–158, 2003.

[Saa88] A. Saalfeld. Conflation: Automated map compilation. *International Journal of Geographical Information Systems*, 2(3):217–228, 1988.

[SB97] Gilbert Strang and Kai Borre. *Linear algebra, geodesy, and GPS*. Wellesley-Cambridge, 1997.

[Sch99] Markus Schneider. Uncertainty management for spatial data in databases: Fuzzy spatial data types. In *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 330–351, London, UK, 1999. Springer-Verlag.

[TN02a] E. Tossebro and M. Nygaard. Abstract and discrete models for uncertain spatiotemporal data. In *SSDBM '02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, page 240, Washington, DC, USA, 2002. IEEE Computer Society.

[TN02b] E. Tossebro and M. Nygaard. An advanced discrete model for uncertain spatial data. In *WAIM '02: Proceedings of the Third International Conference on Advances in Web-Age Information Management*, pages 37–51, London, UK, 2002. Springer-Verlag.

[TN02c] E. Tossebro and M. Nygaard. Uncertainty in spatiotemporal databases. In *ADVIS02: Proceedings of the 2nd International Conference on Advances in Information Systems*, pages 43–53, London, UK, 2002. Springer-Verlag.

[TN03] E. Tossebro and M. Nygaard. A medium complexity discrete model for uncertain spatial data. In *7th Int. Database Engineering and Applications Symposium (IDEAS)*, 2003.

[TN04] E. Tossebro and M. Nygaard. A discrete model for topological relationships between uncertain spatial objects. In *Developments in Spatial Data Handling*, 2004.

University of Verona
Department of Computer Science
Strada Le Grazie, 15
I-37134 Verona
Italy

http://www.di.univr.it