myjournal manuscript No.
(will be inserted by the editor)

A Framework for Integrating Multi-Accuracy Spatial Data in Geographical Applications

Alberto Belussi · Sara Migliorini

Received: date / Accepted: date

Abstract In recent years the integration of spatial data coming from different sources has become a crucial issue for many geographical applications, especially 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 SDI, where continuous and incremental updates occur, will lead very soon to an inconsistent spatial dataset with respect to spatial relations and relative distance among objects. In this paper we address this problem and we propose a framework for representing multi-accuracy spatial databases, based on statistical representation of the objects geometry, together with a method for the incremental and consistent update of the 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. Also spatial relations among objects need to be compared in spatial data integration and we show that they are necessary in order to obtain a correct result in merging objects geometries.

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 problems of spatial data integration has rapidly increased. For instance, many national or regional

A. Belussi
Strada le Grazie 15 - 37134 Verona - Italy
Tel.: +39-045 8027980
Fax: +39-045 8027982
E-mail: alberto.belussi@univr.it

S. Migliorini
Strada le Grazie 15 - 37134 Verona - Italy

geographical agencies, in particular in the European Union, are facing the challenge of integrating in common Spatial Data Infrastructures (SDIs) datasets 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 issue for the following reason. 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. Thus, a certain amount of error in the representation of a spatial object always exists. In literature [20,23,12] the term *accuracy* is considered as a measure of how closely the recorded values represent their true values, and *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 work 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 using a list of coordinates in a reference space) we can observe that: *computer scientists* working with GIS tend to perceive the absolute coordinates as the primary data concerning objects locations and to consider geometric coordinates as deterministic values. The measurements from which these coordinates were obtained are seen as unnecessary data once absolute point locations have been determined and no record is kept about them. 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 calculated coordinates are treated as random variables. The coordinate values are seen simply as a view of the data: the one that best fits the measurements at that time. Moreover, the accuracy of relative geometry is in practice higher than the absolute accuracy; therefore, 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 so in many applications it is not practical. As a consequence in spatial databases only derived coordinates are usually stored without any information about their accuracy or the original measurements from which they come. In literature some papers proposed the introduction of measurements-based cadastral systems [4,11,17] (see also Sec. 2); however, we aim to consider the most frequent case where no details about measurements are available and only coordinates are stored. Having discarded the solution based on measurements management, we still need some aggregated accuracy information in order to deal with spatial data in a correct way, since the derivation of coordinates from observations is a unique but not reversible operation [10]. Moreover, information about accuracy of spatial data should be used in every operation involving these uncertain data; in particular, this 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 thus it is crucial that accuracy becomes a part of spatial data representation at object granularity.

Example 1 Fig. 1 illustrates a typical problem that occurs when an integration is performed by simply 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 this 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, thus 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).

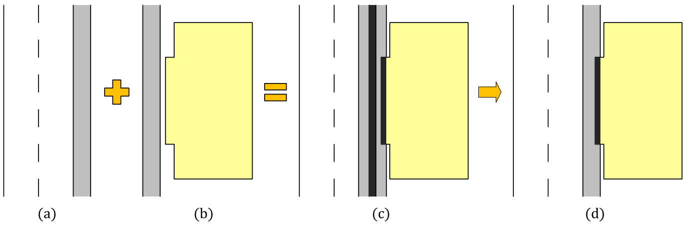


Fig. 1 An example of integration that does not consider the accuracies of the objects to be integrated, but simply replace old objects with new ones.

In this paper we propose a framework for dealing with multi-accuracy spatial databases and treating their integration and update with the aim to solve the problems shown in the previous example. In particular, we suppose that no data about ground measurements are known, but only a set of metadata describing accuracy of absolute positions and accuracy of relative distances have been assigned to each database. This is a very common case in practice, since measurements are not used by GIS applications, that work directly on coordinates often without paying attention to their accuracy. However, at least average errors about absolute positions and relative distances can often be recovered or derived by the cartographic scale of data. More precisely the contribution of this paper is articulated into two points: firstly, in Sec. 3 we define a methodology for computing and representing the accuracy of spatial data starting from the given metadata; secondly, in Sec. 4 we propose an integration procedure, based on the Kalman filter, that considers both the coordinate accuracies 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 introducing the proposed framework for handling multi-accuracy spatial data, we illustrate in the following section 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 [18, 3, 16] 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 [18] they present a method for performing the probabilistic spatial join operation, which, given two uncertain datasets, finds all pairs of polygons whose probability to overlap is larger than a given threshold. In [3, 16] 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 (kNN) 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 introduced by Tossebro et al. in [22–26]. In [23] 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 [25] this model is refined in order to reduce the storage space required and to simplify the computation of the core and support regions. In [24] the authors extend its model with some constructs for representing also temporal uncertainty into a spatial database. Finally, in [26] 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 these works deal with the integration process, they propose a more or less formal model for representing uncertainty and eventually they concentrate on query operations. Conflation techniques [19] 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, no updated quality information are provided for the adjusted dataset.

In [4, 11, 17] the authors introduce the concept of measurement-based GIS as an alternative to the usual notion of coordinate-based GIS. While in the latter systems the stored coordinates values are the primary sources of data and they provide answer to both metric and topological queries; in the proposed kind of GIS only measures between higher-quality points (i.e. control points), parcel boundary measurements and measurements of other objects of interest are stored together with their accuracy information. This solution provides some advantages during the integration process, because any new measure can be easily added to the database, since old or inaccurate measurements can coexists with better values or deleted without difficult. However, any time a query has to be answered or the spatial information has to be visualized, the coordinates of each point have to be derived from measurements. In order to overcome this problem, in [4] the authors propose to store also the obtained coordinates and to periodically process the available measures in order to make coordinates reliable and consistent. As stated in the introduction we consider the more usual case where measurements are not available and only coordinates are stored.

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 [10, 12–14]. These approaches use techniques based on weighted least-squares method to obtain the best fit between the source datasets. The advantage of such approaches is that resultant positions are determined taking into account all the available information, including the positional accuracy of points in both datasets. Moreover, updated quality parameters are generated, enabling detailed quality reporting of the resultant dataset. The integration method proposed here is also based on a least-squares estimation of the new coordinates, but it exploits the Kalman filter to perform an incremental computation of such estimation, namely the integration has not to be performed at once and there is no need to maintain all the previously integrated information for obtaining the final result. In [12–14] the authors consider also the problem of preserving topological relations between objects by representing them as inequalities that are included in the least squares method. In this paper we propose a different approach for preserving topological relations during the integration processes, similarities and differences between the two approaches will be discussed in Sec. 4.3.

In [21] 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 [1] 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. [7]. 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: $f_P(x_P, y_P) : E^2 \to [0, 1]$. 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 $f_P(x_P, y_P)$ by means of a set of parameters that approximate such function. This set of parameters could be very large, moreover visualizing complex probability density functions or using them in query processing could be very difficult and computationally intensive. 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 positions.

Definition 1 (Soft Absolute Position) The absolute position of a point P with probability density function $f_P(x_P, y_P)$, is given 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_P}, μ_{y_P}), where μ_{x_P} and μ_{y_P} are the averages of x_P and y_P 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_P and y_P variables.

$$C_\sigma = \begin{bmatrix} \sigma^2_{x_P} & \sigma_{x_P y_P} \\ \sigma_{y_P x_P} & \sigma^2_{y_P} \end{bmatrix}$$

□

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 positions, namely the maximum granted error between the real coordinates and the measurements, and a validity percentage of that error $F_R(e)$, which is the percentage of cases that have to satisfy this error, for each surveyed area. In [6] the authors illustrate how variance of coordinates can be calculated from these metadata using the circular error formula; in this paper we adopt their approach, as shown in Eq. 1. Since there is no reason for considering different the variance of x from the variance of y, we can suppose that:

$$\sigma_{x_P}^2 = \sigma_{y_P}^2 = \sigma_P^2 = \frac{-e^2}{2 \cdot \log(1 - F_R(e))} \tag{1}$$

Given the variance of a position, the correlation between different positions can be estimated by introducing the covariance between their points coordinates. In this way the correlation is greater for near points and it decreases as distance increases. Given two positions $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$, the variance and covariance values can be represented in a matrix, called C_v, as follows:

$$C_v = \begin{bmatrix} \sigma_P^2 & \sigma_{x_P,y_P} & \sigma_{x_P,x_Q} & \sigma_{x_P,y_Q} \\ \sigma_{y_P,x_P} & \sigma_P^2 & \sigma_{y_P,x_Q} & \sigma_{y_P,y_Q} \\ \sigma_{x_Q,x_P} & \sigma_{x_Q,y_P} & \sigma_Q^2 & \sigma_{x_Q,y_Q} \\ \sigma_{y_Q,x_P} & \sigma_{y_Q,y_P} & \sigma_{y_Q,x_Q} & \sigma_Q^2 \end{bmatrix}$$

This matrix could be calculated only when all measurements collected during surveys are known (as done in [4]). This is not the case considered in this paper, since we suppose to know only some aggregate metadata about the metric accuracy of positions at hand. Under these conditions we must introduce some hypotheses in order to simplify the model and reduce the number of unknown parameters in the matrix.

Definition 2 (Independence hypotheses) Considering surveyed spatial data, the following hypotheses can be reasonable in applications that deal with them, when no detailed information about ground measurements are available:

1. The x and y coordinates of a position P can be considered mutually independent, so the covariance between the x_P and y_P can be set to zero: $\sigma_{x_P,y_P} = \sigma_{x_Q,y_Q} = 0$.
2. We assume that the correlation among point positions has effects only between coordinates of the same axis, i.e. the x (y) coordinate of a position P does not influence the y (x) coordinate of any other point Q, so: $\sigma_{x_P,y_Q} = \sigma_{y_P,x_Q} = 0$
3. The correlation between the x coordinate of P and the x coordinate of Q is equal to the correlation between the y coordinate of P and the y coordinate of Q: $\sigma_{x_P,x_Q} = \sigma_{y_P,y_Q} = c_{PQ}$.

Any other hypotheses leads to inconsistent state of C_v or removes the propagation effect. □

Applying the hypotheses contained in Def. 2 and the covariance property $\sigma_{a,b} = \sigma_{b,a}$, the matrix C_v can be rewritten as follows:

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

where c_{PQ} represents the correlation between the positions P and Q.

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 $\sigma^2_{d_{PQ}}$ is the variance of the relative distance between the two positions P and Q, that can be calculated using Eq. 1 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 $\sigma^2_{d_{PQ}}$ of the relative distance between the two points P and Q and the variance of their coordinates σ^2_P and σ^2_Q, the covariance $\sigma_{x_P,x_Q} = \sigma_{y_P,y_Q} = c_{PQ}$ can be calculated as follows:*

$$c_{PQ} = \frac{\sigma^2_P + \sigma^2_Q - \sigma^2_{d_{PQ}}}{2} \tag{3}$$

Proof - (sketch) Eq. 3 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 distance function between two points. Notice that g is a non-linear function, but it can be easily linearized as $d_{PQ} \simeq J \cdot \overline{v}$, 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. 2, we obtain the thesis. □

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%, then 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. 3:*

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

It follows that every time a value of $\sigma^2_{d_{PQ}}$ greater than this limit is obtained from Eq. 1, it has to be substituted with the value $\sigma^2_P + \sigma^2_Q$. □

Finally, it is easy to prove that with the hypotheses of Def. 2 (in particular the second one) and having imposed the constraint in Obs. 1, the covariance matrix C_v in Eq. 2 is positive-definite. 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.

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

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

- ID is an integer representing an unique identifier for the object.
- CL is the thematic class to which the object belongs, e.g. Building or Road.
- Geo is the geometry of the object, that 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), with no 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 $\emptyset_{geo}.type = null.$ □

Notice that on each object geometry 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 [7] 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].

Many different models for the definition of topological relations have been proposed starting from the well-known *9-intersection model* defined in [7,8]. 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 [5]. 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:

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

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

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.

Considering the value empty (⊘) or not empty (¬⊘) for each intersection, many relations can be distinguished between surfaces, curves and points. In [5], 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. 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 GIS systems topological relations existing 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 topological relations* can be derived, namely topological relations that are not precisely defined.

Claim 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, observed topological relations cannot be considered data subject to measurement error, since we cannot measure them like the width of a building, they can only be true or false. 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 exist 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 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 regards to topological relations three situations may occur: (i) if $|R| = 1$, the relation is known; (ii) if $R = R_U$, the relation is unknown; (iii) if $|R| > 1 \wedge R \neq R_U$, the relation is unknown and could be one of the relations $r \in R$.[2]

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: $r_{soft} \in R$. The integration of two MACS databases can determine the violation of the coherence constraint, we will discuss in detail in Sec. 4.3 how to solve this kind of conflicts.

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

[2] In the following, where there is no ambiguity, a hard topological relation will be denoted simply as topological relation.

topological relations to be stored is $n \times (n-1)/2$, because one topological relation has to be defined between each pair of objects. This could be a large number in real databases, so 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 need to 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 is in general an ellipse around the representative $\underline{P}$. For example, considering the initial state of the matrix C_v for a position P according to Def. 2, then the support for P in this case is a circle with radius $2\sigma_P^2$ for $\alpha \simeq 0.95$.

Given the notion of support for a position P, an index of maximum dispersion α_M can be defined for the whole database: it has to be considered during the computation of the support for each database position. 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 $O = \langle ID, CL, Geo \rangle$ the support of O with respect to α_M (denoted by $Supp(O, \alpha_M)$) can be approximated by considering the smallest buffer region of $O.Geo.rep$[3] that contains the support of all its defining positions $O.Geo.pos$. The real position of an object cannot be outside its support. □

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. 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 other ones 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.

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).

Given the definition of soft and hard data we can define a MACS database as follows.

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

[3] 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.

- *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 $\underline{\mathrm{P}}$ the following tuple is stored: $\langle ID_P, x_P, y_P\rangle$, 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 $\langle ID, CL, Geo\rangle$ (see Def. 3) belonging to the classes of *TY* and whose geometry is described through the positions in *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 $Supp_{DB}$ is the region representing the support of the database, which is obtained as the union of the objects supports.

Notice 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 their boundary). □

We propose different methods for storing C_{DB} that can be applied in different states of the database. Initially the matrix can be generated starting from two metadata describing the metric quality of the whole database by applying the procedure shown in Sec. 3.1. We observe that in practice the error of relative distance e_d is usually considered as a function of the distance d, for instance it can be a function like: $f_{e_d}(d) = (0.60 + d/1000)$ for $d \leq 600m$ and $f_{e_d}(d) = 1.20$ for $d > 600m$. In this example the computed covariance (see Eq. 3) is greater than 0 only for points having a relative distance less than $600m$. On the contrary, error of absolute positions is usually constant. Therefore, initially only these metadata have to be stored: namely we store the pair $(e, FR(e))$ and the pair $(f_{e_d}(d), FR(e_d))$ (see Sec. 3.1), representing the error of the absolute positions and the error of the relative distance among positions, respectively. The initialization step could be more complex if several metadata about metric quality are available, for instance we could have different metadata in different regions partitioning $Supp_{DB}$. In this case we store the region together with its metadata: for example, $(e, FR(e), R_1)$ if the metadata $(e, FR(e))$ is valid in R_1. After the integration with another database we might need to store some values of C_{DB}, in particular those values that have a significant difference (greater than a given threshold) with respect to the ones obtained from metadata. We call this matrix C^{δ}_{DB} and we store it together with the metadata. Other optimizations can be applied; for instance, the covariance values of the matrix C^{δ}_{DB} can be approximated by storing them only for a subset of position pairs; this subset can be determined for example by using a Delaunay triangulation. An alternative approach could identify regions (clusters) with positions having homogeneous variance (covariance) values and replace the portion of C^{δ}_{DB} regarding these points with a set of metadata of the form: $\{(e_1, FR(e_1), R_1), ..., (e_k, FR(e_k), R_k)\}$ $(\{(f_{e_{d_1}}, FR(e_{d_1}), R_1), ..., (f_{e_{d_k}}, FR(e_{d_k}), R_k)\})$.

Example 2 (Example of MACS database) Let us consider the database presented in Fig. 1(a), denoted here as DB^1_m. Supposing that for DB^1_m the error e for the absolute position is $0.8m$ with a percentage of validity of 95%, and the error e_d for the relative distance is $0.6m$ 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* 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.

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

3.3 MACS database accuracy estimators

In order to evaluate the overall accuracy of a MACS database, we introduce an index of metric accuracy and an index of certainty for logic observations. We choose to give an estimation of certainty of 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_m, the metric accuracy of its absolute position is defined as the inverse of its variance. Since according to Eq. 2 the variance of the x and y coordinates of a point is the same and, as we will see in next sections, remains the same also after the integration procedure, the metric accuracy of the position P is defined as: $acc_M(P) = 1/\sigma^2_P$.

The *average global accuracy estimation* of a MACS database DB_m concerning the metric observations can be computed as:

$$acc_M(DB_m) = \frac{\sum_{P_i \in DB_m.DB} acc_M(P_i)}{|DB_m.DB|}$$

Similarly, we can observe that the certainty of a set of topological relations R defined between two objects O_1 and O_2 can be estimated as: $acc_T(R) = (|R_U| - |R|)/((|R_U| - 1) \cdot |R|)$. Considering the reference set of topological relations proposed in Sec. 3.2, we obtain: $acc_T(R) = (7 - |R|)/(6 \cdot |R|)$. Therefore, the certainty is the highest when $|R| = 1$, namely when the relation is known ($acc_T(R) = 1$), and it is the lowest when $R = R_U$, namely when the relation is unknown ($acc_T(R) = 0$).

The *average global certainty estimation* of a MACS database DB_m concerning the logic observation can be computed as follows:

$$acc_T(DB_m) = \frac{|DB_m.OBJ|^2 - |DB_m.REL| + \sum_{R_i \in DB_m.REL} acc_T(R_i)}{|DB_m.OBJ|^2}$$

Each known topological relation (i.e. not explicitly stored in REL) has a unit certainty value, so the first term $|DB_m.OBJ|^2 - |DB_m.REL|$ calculates the overall certainty of all known relations. To this value the certainty of all unknown topological relations is added ($\sum_{R_i \in DB_m.REL} acc_T(R_i)$). This sum is normalized with respect to the total number of possible relations ($|DB_m.OBJ|^2$), 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 databases to be integrated can be completely different or can share absolute positions and/or objects and/or relations. More specifically, different application scenarios may occur during 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 observations 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 $DB^1_m = (DB_1, C_{DB_1}, TY_1, OBJ_1, REL_1, \alpha_M, Supp_{DB_1})$ and $DB^2_m = (DB_2, C_{DB_2}, TY_2, OBJ_2, REL_2, \alpha_M, Supp_{DB_2})$ their integration produces a new database $DB^3_m = (DB_3, C_{DB_3}, TY_3, OBJ_3, REL_3, \alpha_M, Supp_{DB_3})$ whose components can be obtained by applying different operations to the corresponding components of DB^1_m and DB^2_m, depending on the interaction that exists between them, as reported in Table 3, in particular:

$$\begin{aligned} DB_3 &= \mathit{metricPosInt}(DB_1, DB_2, C_{DB_1}, C_{DB_2}) \\ C_{DB_3} &= \mathit{metricVarInt}(C_{DB_1}, C_{DB_2}) \\ TY_3 &= TY_1 \oplus_{ty} TY_2 \\ OBJ_3 &= OBJ_1 \oplus_{obj} OBJ_2 \\ REL_3 &= \mathit{logicRelInt}(REL_1, OBJ_1, REL_2, OBJ_2) \end{aligned}$$

□

Notice that, in Table 3 some combinations are not admissible and are not shown, since the following conditions have to be satisfied:

$$\begin{aligned} OBJ_1.ID \cap OBJ_2.ID \neq \emptyset &\implies ext(REL_1, OBJ_1) \cap ext(REL_2, OBJ_2) \neq \emptyset \\ OBJ_1.ID \cap OBJ_2.ID \neq \emptyset &\implies TY_1 \cap TY_2 \neq \emptyset \end{aligned}$$

The preliminary opration 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 an 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.

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

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

The simplest integration tasks are those regarding classes and objects. Indeed, the integration of 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:

$$
\begin{aligned}
OBJ_1 \oplus_{obj}\ & OBJ_2 = \\
& \{o \mid (o \in OBJ_1 \wedge o.ID \notin OBJ_2.ID)\ \vee (o \in OBJ_2 \wedge o.ID \notin OBJ_1.ID)\}\ \cup \\
& \{objPosInt(o_1, o_2) \mid o_1 \in OBJ_1 \wedge o_2 \in OBJ_2 \wedge o_1.ID = o_2.ID\} \qquad (4)
\end{aligned}
$$

where $objPosInt(o_1, o_2)$ is the procedure that identifies which positions have to be integrated and stored in the final database DB^3_m as representatives for the object with the same *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 is necessary.

The next subsections are organized as follows, first the integration of the selected positions (metric observations) is considered in Sec. 4.1; in particular, a statistical method for computing the functions $metricPosInt(DB_1, DB_2, C_{DB_1}, C_{DB_2})$ and $metricVarInt(C_{DB_1}, C_{DB_2})$ is presented. In Sec. 4.2 we concentrate on the problem of integrating topological relations (logic observations); more specifically, a method for computing the function $logicRelInt(R_1, R_2)$ 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 in two MACS databases. This method is denoted here as $metricPosInt_{kalman}$ $(DB_1, DB_2, C_{DB_1}, C_{DB_2})$, where DB_1 and DB_2 are the set of positions contained in the two databases, while C_{DB_1} and C_{DB_2} are the corresponding dispersion index matrices. This method is based on an application of the Kalman filter [15] to the vectors of coordinates, containing the representative of the positions that have to be integrated, and the matrices of the variance-covariance estimates for such positions.

The use of the Kalman filter for performing the integration has the following important advantage: 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 [21] 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 with the current estimate $\hat{x}_{k|k}$.

Notice that, in order to effectively integrate two databases, they should share a common area; otherwise, there is no possibility to define a real correlation between them and no adjustments 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. Nevertheless, the proposed method is able to deal with all the cases 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:*

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

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

where the matrix C_{zero} contains only zeros.

Proof – This result is due to Obs. 1 and the hypotheses in Def. 2, since no information is available about the relative distance among the objects of the databases to be integrated. □

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, we need to prepare the coordinates vectors, one for each database to be integrated, and the corresponding matrices of variance-covariance estimates that will be used by the Kalman filter. Notice that each vector (matrix) should contain coordinates (variance-covariance values) regarding the whole set of objects the resulting MACS database will contain. We denote the coordinates vectors as V_{DB_1}, V_{DB_2} and the variance-covariance matrices as C'_{DB_1} and C'_{DB_2}. They 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. (4)), then*

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

- *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 replace 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, $DB_i \setminus_{ID} DB_j = \{p \mid p \in DB_i \wedge p.ID \notin DB_j.ID\}$, $DB_i \cap_{ID} DB_j = \{p \mid p \in DB_i \wedge p.ID \in DB_j.ID\}$[4] *and $\Pi_{a,b}(C)$ computes*

[4] Notice that $\cap_{ID}$ is not commutative

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 replace 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 filter is straightforward.

Method 1 (Position Integration (Kalman filter)) *Given the vectors V_{DB_1}, V_{DB_2} and the matrix C'_{DB_1}, C'_{DB_2} (see Obs. 3) the Kalman filter is applied as follows:*

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

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

$$K = ((C'_{DB_1})^{-1} + (C'_{DB_2})^{-1})^{-1} \cdot (C'_{DB_2})^{-1} \tag{5}$$

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.

$$V_{DB_3} = V_{DB_1} + K \cdot (V_{DB_2} - V_{DB_1}) \tag{6}$$

□

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

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:

$$C_{DB_3} = (I - K) \cdot C'_{DB_1} \cdot (I - K)^T + K \cdot C'_{DB_2} \cdot K^T \tag{7}$$

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

Let us denote with $c^1_{a,b}$, $c^2_{a,b}$, $c^3_{a,b}$ and $k_{a,b}$ the coefficients of the matrices C_{DB_1}, C_{DB_2}, C_{DB_3} and K, respectively, in row a and column b. It is easy to prove that the following properties holds:

- If for a certain position P it holds that $c^1_{x_P,x_P} = c^1_{y_P,y_P}$ and $c^2_{x_P,x_P} = c^2_{y_P,y_P}$, then it follows that $c^3_{x_P,x_P} = c^3_{y_P,y_P}$ and $k_{x_P,x_P} = k_{y_P,y_P}$.
- If for a certain position P it holds that $c^1_{x_P,y_P} = c^1_{y_P,x_P} = 0 = c^2_{x_P,y_P} = c^2_{y_P,x_P}$, then it follows that $c^3_{x_P,y_P} = c^3_{y_P,x_P} = 0$ and $k_{x_P,y_P} = k_{y_P,x_P} = 0$.
- If for a certain pair of positions P and Q, it holds that $c^1_{x_P,x_Q} = c^1_{y_P,y_Q}$ and $c^2_{x_P,x_Q} = c^2_{y_P,y_Q}$, then it follows that $c^3_{x_P,x_Q} = c^3_{y_P,y_Q}$ and $k_{x_P,x_Q} = k_{y_P,y_Q}$.
- If for a certain pair of positions P and Q, it holds that $c^1_{x_P,y_Q} = c^1_{y_Q,x_P} = c^1_{y_P,x_Q} = c^1_{x_Q,y_P} = 0 = c^2_{x_P,y_P} = c^2_{y_P,x_P} = c^2_{y_P,x_Q} = c^2_{x_Q,y_P}$, then it follows that $c^3_{x_P,y_Q} = c^3_{y_Q,x_P} = c^3_{y_P,x_Q} = c^3_{x_Q,y_P} = 0 = k_{x_P,y_Q} = k_{y_Q,x_P} = k_{y_P,x_Q} = k_{x_Q,y_P}$.

These properties confirms that the initial configuration of the variance-covariance matrix (see Eq. 2) is preserved by the proposed integration methods.

Notice that the effectiveness of a least square-based method can be compromised by the presence of blunders. A blunder is an erroneous observation that is clearly in contrast with the other available observations. In the integration context, blunders influence the point matching phase of the two source databases. Several blunder detection techniques have been proposed [?]; however, in this paper we assume that the quality of the considered information is ensured by the data provider that is responsible for performing a correct point matching of the source databases, thus we can safely abstract from this problem.

It is clear that in real situation a 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 distance threshold for covariance values has been defined, so that covariance values are different from zero and 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 correlated to the new integrated ones, namely whose positions that are within the distance specified by the error function for the relative distances, since for the other positions the covariance would be equal to zero and thus no integration effects would spread on them.

4.2 Integrating Logic Observations

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

As regards the integration of logic observations, a significant case occurs when the two databases share at least one object. Anyway, the proposed method is able to handle any possible case; indeed, different operations are necessary according to the rate of objects sharing. 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 objects O_1 and O_2 respectively, where $O_1.ID \cap O_2.ID = \emptyset$, the following function can be*

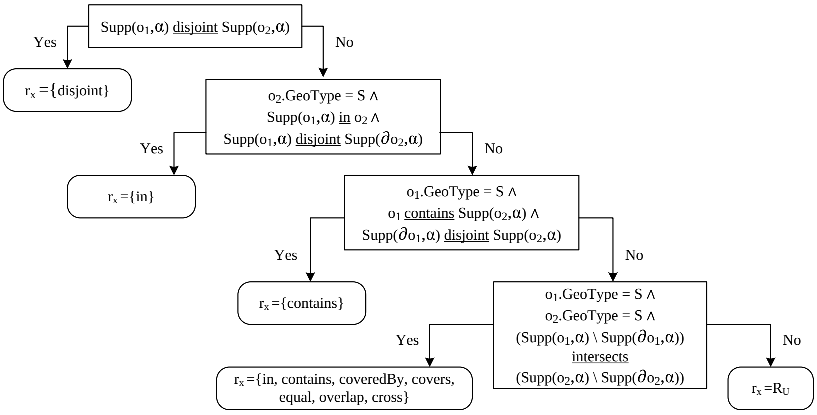

Supp(o1,α) disjoint Supp(o2,α)
Yes
No
rx ={disjoint}
o2.GeoType = S ∧
Supp(o1,α) in o2 ∧
Supp(o1,α) disjoint Supp(∂o2,α)
Yes
No
rx ={in}
o1.GeoType = S ∧
o1 contains Supp(o2,α) ∧
Supp(∂o1,α) disjoint Supp(o2,α)
Yes
No
rx ={contains}
o1.GeoType = S ∧
o2.GeoType = S ∧
(Supp(o1,α) \ Supp(∂o1,α))
intersects
(Supp(o2,α) \ Supp(∂o2,α))
Yes
No
rx ={in, contains, coveredBy, covers, equal, overlap, cross}
rx =RU

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

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:*

$$topFromSupp(O_1, O_2) = \{\langle o_1, o_2, r_x\rangle \mid (o_1, o_2) \in O_1 \times O_2 \wedge r_x = f_{supp}(o_1, o_2)\}$$

where $f_{supp}(o_1, o_2)$ *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 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_1. The third conditional block shows a situation that is the inverse of the previous one. Finally, the last conditional block says that, if the supports of two surfaces intersect without considering the support of their interior, the surfaces certainly have intersecting interiors, thus the existing relation between them can be only one among: in, contains, covers, coveredBy, equal or overlap. □

Method 2 (Relation integration) *Given two distinct MACS databases* DB^1_m*,* DB^2_m*, the integration of the sets of topological relations (or logic observations) that are known in each of them is represented by the function* $logicRelInt(REL_1, OBJ_1, REL_2, OBJ_2)$*. In order to obtain this result we first compute the complete set of relations known by* DB^1_m *(*DB^2_m*), 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 databases to be integrated* DB^1_m *and* DB^2_m*:*

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

where $topFromSupp(OBJ_1, OBJ_2)$ *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 databases to be integrated, thus the function works differently:*

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

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 $\mathit{mergeTopRel}(R_1, O_1, R_2, O_2)$ *is defined as follows:*

$$\begin{aligned} \mathit{mergeTopRel}(R_1, O_1, R_2, O_2) = \{&\langle o_1, o_2, r\rangle \mid o_1 \in O_1 \wedge o_2 \in O_2 \wedge \\ &\langle o_1, o_2, r_1\rangle \in R_1 \wedge \langle o_1, o_2, r_2\rangle \in R_2 \wedge r = r_1 \cap r_2)\} \end{aligned} \quad (8)$$

The result of $\mathit{logicRelInt}(REL_1, OBJ_1, REL_2, OBJ_2) = REL_3$ *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 databases to be integrated and have to be solved by human intervention.

The human intervention is necessary whenever logic observations contained in the source databases are discordant. 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, some automatic procedures can be implemented in order to convert the inconsistency into a loss of certainty. For this purpose we consider the proximity relationship among topological relations, first introduced in [9] for the definition of conceptual neighborhoods starting from the the 9-intersection matrices. This definition has been extended in [2] 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 paper we adopt the same approach for defining, given a topological relation REL_1 between specific object types (e.g. between surfaces), the set of relations that are near to it. We say that a topological relation REL_1 is *near* to another relation REL_2, if REL_2 is characterized by a matrix with the minimum distance (variation), with respect to other relations, from 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.

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, then the resulting relation becomes the disjunction of the original ones. Formally, this result can be obtained by replacing the intersection $r_1 \cap r_2$ in Eq. 8 with $near(r_1, o_1.\mathit{Geo.type}, o_2.\mathit{Geo.type}) \cap near(r_2, o_1.\mathit{Geo.type}, o_2.\mathit{Geo.Type})$, 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.

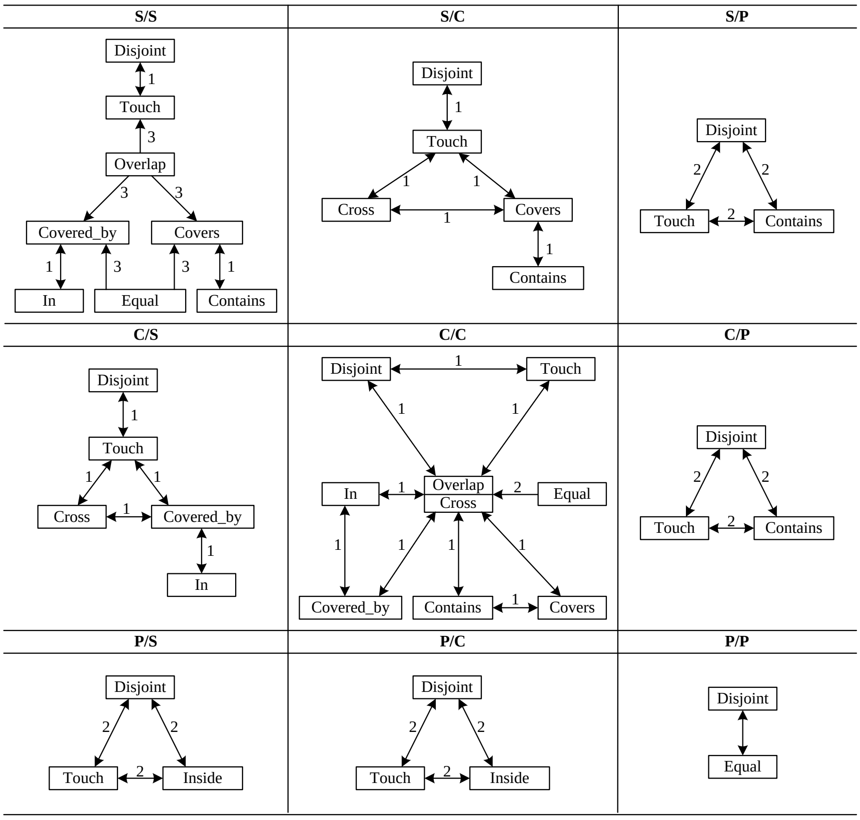


Fig. 3 Proximity between topological relations classified on the basis of the type of the involved objects. Let us notice that for not cluttering the diagram, the relations *Overlap* and *Cross* between two curves have been collapsed into a unique box because they have the same distance from the other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

4.3 Integrating Metric and Logic Observations Together

The complete integration of two MACS databases requires to combine metric and logic observations together. In particular, in Sec. 3.2 we have introduced 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 a topological relation is known, it can be derived directly from the geometries of the objects representatives without additional information.

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_m. This means that for each pair of objects of OBJ_3 we need to compute the soft topological relation between them, 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

r_{soft} changes and becomes one of the relations of R. Indeed, we always suppose that logic observations have higher 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* → *touch*, require that two distinct positions of the objects involved in the 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.

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.

Method 3 (Alignment of positions with respect to logic observations) *Given an integrated MACS databases DB_m^3 and the initial databases DB_m^1 and DB_m^2 the alignment of positions with respect to logic observations is an iterative process that is executed until the following condition holds:*

$$\{(o_1, o_2) \mid (o_1, o_2) \in OBJ_3 \times OBJ_3 \wedge o_1 r_{soft} o_2 \wedge \langle o_1, o_2, R \rangle \in REL_3 \wedge r_{soft} \notin R\} = \emptyset$$

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

1. *for each violation of consistency between a pair of objects (o_1, o_2), the necessary relation transition $r_A \rightarrow r_B$ is identified;*

2. *for each relation transition its applicability is evaluated; in particular, some transition are not admitted a priori, some others requires operations that, in specific cases, could not be applied (see Tables 4-6 in this section and Tables 7-15 in Appendix);*
3. *for each relation transition that is not applicable, since its preconditions are not satisfied, the user intervention is required;*
4. *for each relation transition $r_A \rightarrow r_B$ that is applicable and such that $o_1\ r_{soft}\ o_2$ in DB_m^i ($i \in \{1,2\}$) and $r_{soft} = r_B$, we augment the accuracy of the relative distance among all the positions pairs $(P_i, Q_i) \in o_1.Geo.pos \times o_2.Geo.pos$ having intersecting supports by setting the covariance to the value $(\sigma_P^2 + \sigma_Q^2)/2$ (see Eq.3) in the corresponding matrix C_{DB_i}. In this way the accuracy of the relative distance (i.e. the covariance) between these two objects P and Q becomes maximum, the two objects now constitute a rigid body and move accordingly without changing the relative position of their points. Then we repeat the computation of $DB_3 = metricPosInt_{kalman}(DB_1, DB_2, C_{DB_1}, C_{DB_2})$.*
5. *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: $DB_{final} = metricPosInt_{kalman}(DB_3, DB_3', C_{DB_3}, C_{DB_3'})$.*

□

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).

Notice that, in Tables 7-15 some allowed transitions involves 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 that the positions of objects involved into a particular topological relation have to become a rigid body that can move in

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 $a \overset{in}{\leftarrow\rightrightarrows} b$
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 $a \overset{out}{\leftarrow\rightrightarrows} b$

Table 6 Positions switching operations.

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 proposed in phase 4 and in the operations eventually applied in phase 5.

Our approach differs from to the one presented in [12, 13] for several reasons; 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 contain 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 [12][13], 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: in practice very few relations are violated after the integration process.

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

Theorem 1 (Operations for disjoint transitions) *Let us consider Table 7 showing the allowed transitions starting from the disjoint relation. Each column, representing a given target relation $REL_{*,*}$, reports for each types pair t_1, t_2 the operations that represent a sufficient condition in order 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*) → (*a touch b*), for types 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 are locally modified, so that after the modification a and b share a position. As a consequence, the intersection $\partial a \cap \partial b$[5] 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 types 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 types 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 process proposed in Sec. 4. In particular, we start by discussing the central role covered by the accuracy of each measure during the integration process, showing that the final position of a location depends not only on the integrated measures, but also on their accuracy and their correlation with near positions. Then we state that the accuracy of the integrated measures 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, respectively. In order to demonstrate these properties, we first analyze the trend of the coefficients of the Kalman matrix in relation to the different accuracies of the two source databases. Given two MACS databases DB_m^1 and DB_m^2 that have to be integrated, the coefficients of the Kalman matrix associate to each absolute or relative measure in DB_m^2 a value proportional to its accuracy and normalized with respect to the overall accuracy of the two source databases. In particular, the coefficients of the Kalman matrix assume a value as follows:

- The coefficients $k_{x_P,x_P} = k_{y_P,y_P}$ related to the variance of a position P have a value between 0 and 1.

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

- The coefficients $k_{x_P,x_Q} = k_{y_P,y_Q}$ for $Q \neq P$ related to the covariance between two different positions P and Q have a value between -1 and 1.

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

[5] Here we use the notation adopted by the 9-intersection model [7], that is presented in section 3.2.

- The coefficients k_{x_P,y_P} and k_{x_P,y_Q} corresponding to the covariance between the x and y coordinates of a same position P, and the coefficients k_{y_P,x_P}, and k_{y_Q,x_P} for $Q \neq P$ corresponding to the covariance between the x and y coordinate of two distinct positions P and Q are zero.

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

Property 1 *Given two MACS databases DB^1_m and DB^2_m that have to be integrated, the shift of a position P from its location in DB^1_m increases if the accuracy of P in DB^2_m is greater than the accuracy of P in DB^1_m.*

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_m is obtained from the Eq. 6 as:

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

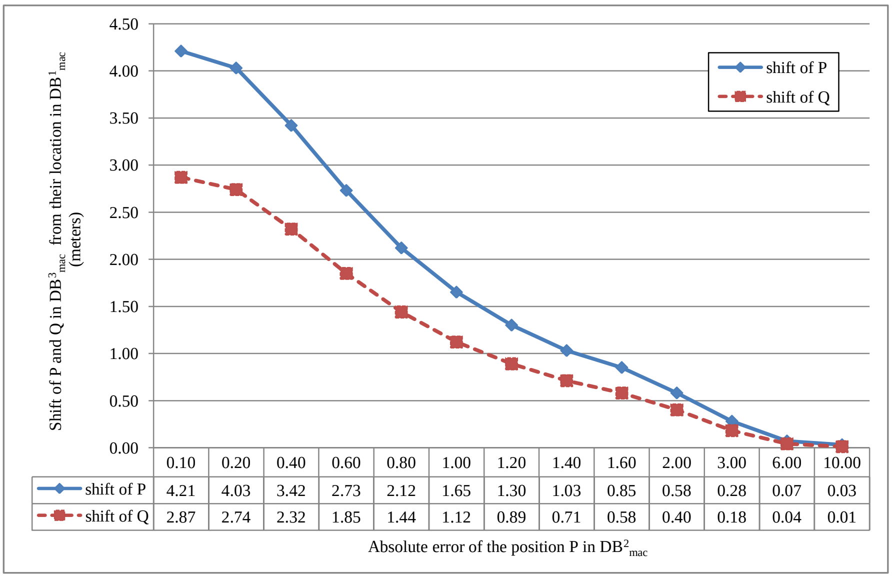
Let us suppose for simplicity that inside the two source databases there are only two positions $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$. The shift of the integrated x coordinate of the position P, denoted as x^3_P, from its original value in DB^1_m becomes:

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

where $k_{i,j}$ is the coefficient of the Kalman matrix in row i and column j. Independently from the measurements contained in DB^2_m (x^2_P and x^2_Q), the shift of x^3_P from the value x^1_P directly depends upon the coefficient k_{x_P,x_P} and k_{x_P,x_Q} of the Kalman matrix. The trend of the Kalman matrix coefficients states that the more the accuracy of the position P in DB^2_m increases with respect to the accuracy of the same position in DB^1_m, the more the value of the coefficients k_{x_P,x_P} and k_{x_P,x_Q} tends to one, determining a greater shift of x^3_p that can eventually become equal to x^2_P. Notice that the shift of P is due not only to a direct update of its measure in DB^2_m, 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_m contains two positions $P = (100, 100)$ and $Q = (123, 123)$ that have both an absolute accuracy e of $0.8m$ (with $F_R(e) = 95\%$), while their relative distance has an accuracy of $0.6m$ (with $F_R(e_d) = 95\%$). Moreover, DB^2_m contains another measure for $P = (103, 103)$ that has to be integrated with the one contained in DB^1_m. We perform the integration between the measures of the two source databases varying the error $e(P)$ of P in DB^2_m and we analyze the different shift of P and Q in DB^3_m from their positions in DB^1_m. The results of this test are reported in the graph of Fig. 4. The graph clearly illustrates that greater is the accuracy of x_P in DB_2 (smaller is its circular error), greater 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 only due to the propagation of the P integration.

Property 2 *Given the MACS database DB^3_m obtained by integrating two source MACS databases DB^1_m and DB^2_m, the accuracy of each integrated measure in DB^3_m 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.*



	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
shift of P	4.21	4.03	3.42	2.73	2.12	1.65	1.30	1.03	0.85	0.58	0.28	0.07	0.03
shift of Q	2.87	2.74	2.32	1.85	1.44	1.12	0.89	0.71	0.58	0.40	0.18	0.04	0.01

Fig. 4 Shift of the positions P and Q with respect to their original measures in DB^1_m, considering different absolute error $e(P)$ for P in DB^2_m.

Proof – The metric accuracy of a position P is defined in Sec. 3.3 and it inversely depends on the positions variance. The variance for the integrated position P in DB^3_m is computed using the Eq. 7 as:

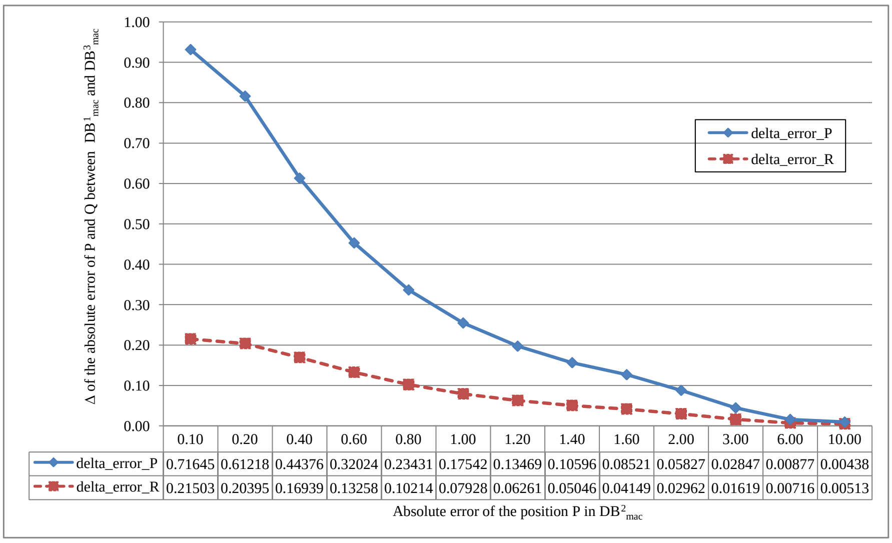
$$C_{DB_3} = (I - K) \cdot C_{DB_1} \cdot (I - K)^T + K \cdot C_{DB_2} \cdot K^T$$

Let us suppose that DB^2_m contains a very accurate measure for P, then as stated above the coefficient k_{x_P,x_P} (or equivalently k_{y_P,y_P}) of the Kalman matrix has a value near to one. From this, it follows that the resulting variance value in C_{DB_3} is very close to (at most coincides with) the element contained in C_{DB_2}, namely to the most accurate one. Conversely, if DB^1_m contains a very accurate measure for P, then the coefficient k_{x_P,x_P} of the Kalman matrix has a value near to zero and the variance value in C_{DB_3} for P is very close 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_{x_P,x_P} of the Kalman matrix contains a positive but smaller than one value. This value multiplied with the elements of the original matrices produces a value that is 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 increases. □

Example 4 Let us consider again the two MACS databases in Ex. 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 are reported in the graph of Fig. 5, considering different values of absolute error $e_2(P)$ for the position P in DB^2_m. We can notice that as P becomes more accurate, the error of the integrated measures decreases. Moreover, if the error $e_2(P)$ is equal to the error $e_1(P)$

of P in DB_1 ($0.80m$), the integrated measure has an error that is smaller than 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, it has not effect during the integration process also as regards to the error of the integrated measure, indeed as $e_2(P)$ increases, the resulting error for the integrated measure settles to a value near the original error in DB^1_m ($0.80m$).

From this property and the definition of average global accuracy of metric observations, it directly derives that also the average global accuracy of metric observations of an integrated MACS database is always greater than or equal to the maximum average global accuracy of metric observations of the source databases.



	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	0.71645	0.61218	0.44376	0.32024	0.23431	0.17542	0.13469	0.10596	0.08521	0.05827	0.02847	0.00877	0.00438
delta_error_R	0.21503	0.20395	0.16939	0.13258	0.10214	0.07928	0.06261	0.05046	0.04149	0.02962	0.01619	0.00716	0.00513

Fig. 5 Variation of the absolute error for the integrated positions P and Q in DB^3_m with respect to the value in DB^1_m, considering different absolute error for P in DB_2.

Property 3 *Given a MACS database DB^3_m obtained by integrating two source MACS databases DB^1_m and DB^2_m, 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. 4. Discarding the optimization mentioned at the end of Sec. 4.2, given two disjunction of topological relations R^1 and R^2, their integration always produces a set of relations R^3 whose cardinality is smaller than the cardinality of both the original ones, or equal to the smallest ones ($|R^3| \leq min(|R^1|, |R^2|)$). Therefore, putting this new cardinality into the certainty formula, we obtain a certainty index that is equal to the greater one or is greater than both the original ones. □

From this property and the definition of average global certainty of logic observations, it directly derives that also the average global certainty of logic observations of an integrated MACS database is always greater than or equal to the maximum average global certainty of metric observations of the source databases.

The presented properties allows one to conclude 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 together 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.

References

1. R. B. Altman. A Probabilistic Algorithm for Calculating Structure: Borrowing from Simulated Annealing. In *9th Annual Conference on Uncertainty in Artificial Intelligence*, Washington, D. C., 1993. Morgan Kaufman.
2. Alberto Belussi, Barbara Catania, and Paola Podestà. Towards Topological Consistency and Similarity of Multiresolution Geographical Maps. In *13th Annual ACM International Workshop on Geographic Information Systems*, pages 220–229, New York, NY, USA, 2005. ACM.
3. B. Bhanu, R. Li, C. Ravishankar, M. Kurth, and J. Ni. Indexing Structure for Handling Uncertain Spatial Data. In *6th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, 2004.
4. T. B. Buyong, B.M Taher, Frank A. U., and W. Kuhn. A Conceptual Model of Measurement-Based Multipurpose Cadastral Systems. *Journal of the Urban and Regional Information Systems Ass. URISA*, (2):35–49, 1991.
5. Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In *3rd International Symposium on Advances in Spatial Databases (SSD'93)*, pages 277–295, 1993.
6. 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.
7. Ma. J. Egenhofer and Ra. D. Franzosa. Point-set Topological Spatial Relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.
8. Max J. Egenhofer and Robert D. Franzosa. On the Equivalence of Topological Relations. *International Journal of Geographical Information Systems*, 9(2):133–152, 1995.
9. 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.
10. 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.

11. M.F. Goodchild. Measurement-based GIS. In W. Shi, P.F. Fisher, , and M.F. Goodchild, editors, *Spatial Data Quality*, pages 5–17. Taylor and Francis, 2002.
12. S. Hope. *Integration of Vector Datasets*. PhD thesis, Department of Geomatics, University of Melbourne, July 2008.
13. S. Hope and A. Kealy. Using Topological Relationships to Inform a Data Integration Process. *Transactions in GIS*, 12(2):267–283, 2008.
14. 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.
15. R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal Of Basic Engineering*, 82:35–45, 1960.
16. 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.
17. G. Navratil, M. Franz, and E. Pontikakis. Measurement-Based GIS Revisited. In *7th AGILE Conference on Geographic Information Science*, pages 771–775, 2004.
18. J. Ni, C. V.. Ravishankar, and B. Bhanu. Probabilistic Spatial Database Operations. In *Advances in Spatial and Temporal Databases*, pages 140–158, 2003.
19. A. Saalfeld. Conflation: Automated Map Compilation. *International Journal of Geographical Information Systems*, 2(3):217–228, 1988.
20. Markus Schneider. Uncertainty Management for Spatial Data in Databases: Fuzzy Spatial Data Types. In *6th International Symposium on Advances in Spatial Databases*, pages 330–351, London, UK, 1999. Springer-Verlag.
21. Gilbert Strang and Kai Borre.
22. E. Tossebro and M. Nygaard. Abstract and Discrete Models for Uncertain Spatiotemporal Data. In *14th International Conference on Scientific and Statistical Database Management SSDBM02*, page 240, 2002.
23. E. Tossebro and M. Nygaard. An Advanced Discrete Model for Uncertain Spatial Data. In *3th International Conference on Advances in Web-Age Information Management (WAIM'02)*, pages 37–51, 2002.
24. E. Tossebro and M. Nygaard. Uncertainty in Spatiotemporal Databases. In *2nd International Conference on Advances in Information Systems (ADVIS02)*, pages 43–53, 2002.
25. E. Tossebro and M. Nygaard. A Medium Complexity Discrete Model for Uncertain Spatial Data. In *7th Int. Database Engineering and Applications Symposium (IDEAS)*, 2003.
26. E. Tossebro and M. Nygaard. A Discrete Model for Topological Relationships between Uncertain Spatial Objects. In *Developments in Spatial Data Handling*, 2004.

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* → *rel*.

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

Table 8 Transition between topological relations: case *equal* → *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) $a \overset{in_L}{\longleftrightarrow} b$	NA	NA
$t_{S,C}$	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$	ND	NA	ND	(req. 1) $a \overset{cr}{\longleftrightarrow} b$	ND	ND	(req. 1) $a \overset{in_L}{\longleftrightarrow} b$
$t_{S,P}$	(2) $a \overset{out}{\leftleftarrows\rightrightarrows} b$	ND	(2) $a \overset{in_L}{\longleftrightarrow} b$	ND	ND	ND	ND	
$t_{C,S}$	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$	NA	ND	ND	(req. 1) $a \overset{cr}{\longleftrightarrow} b$	ND	(req. 1) $b \overset{in_L}{\longleftrightarrow} a$	ND
$t_{C,C}$	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$	NA	NA	NA	(1) $a \overset{cr}{\longleftrightarrow} b$	(1) $a \rightarrow\leftarrow_2 b$	NA	NA
$t_{C,P}$	(2) $a \overset{out}{\leftleftarrows\rightrightarrows} b$	ND	(2) $a \overset{in_L}{\longleftrightarrow} b$	ND	ND	ND	ND	ND

Table 9 Transition between topological relations: case *touch* → *rel*.

$a\ i_{*,*}\ b \rightarrow a\ rel\ b$								
$i_{*,*}$	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* → *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* → *rel*.

$a\ r_{*,*}\ b \rightarrow a\ rel\ b$								
$r_{*,*}$	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* → *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 $\overset{cr}{\leftarrow\rightarrow}$ b	NA	NA

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

$a\ b_{*,*}\ b \rightarrow a\ rel\ b$								
$b_{*,*}$	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) $a \overset{out}{\leftarrow\rightarrow} b$	NA
$b_{C,S}$	NA	(1) $b \rightarrow\leftleftarrows a$	(1) $a \overset{in_L}{\leftleftarrows\rightrightarrows} b$	ND	ND	$a \overset{out}{\leftarrow\rightarrow} b$	ND	ND
$b_{C,C}$	NA	NA	(1) $\partial a \overset{in_L}{\leftleftarrows\rightrightarrows} b$	NA	NA	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a^\circ \rightarrow\leftarrow_1 b^\circ$	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a \rightarrow\leftarrow_2 b$	NA

Table 14 Transition between topological relations: case *coveredBy* → *rel*.

$a\ v_{*,*}\ b \rightarrow a\ rel\ b$								
$v_{*,*}$	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) $a \overset{out}{\leftarrow\rightarrow} b$	NA
$v_{S,C}$	NA	(1) $a \rightarrow\leftleftarrows b$	ND	(1) $b \overset{in_L}{\leftleftarrows\rightrightarrows} a$	ND	(1) $a \overset{out}{\leftarrow\rightarrow} b$	ND	ND
$v_{C,C}$	NA	NA	NA	(1) $\partial b \overset{in_L}{\leftleftarrows\rightrightarrows} a$	NA	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a^\circ \rightarrow\leftarrow_1 b^\circ$	(1) $a \overset{out}{\leftleftarrows\rightrightarrows} b$ $a \rightarrow\leftarrow_2 b$	NA

Table 15 Transition between topological relations: case *covers* → *rel*.