ISPRS Journal of Photogrammetry and Remote Sensing 92 (2014) 147–163

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Data matching of building polygons at multiple map scales improved by contextual information and relaxation

Xiang Zhang [a,*], Tinghua Ai [a], Jantien Stoter [b], Xi Zhao [c]

[a] School of Resource and Environmental Sciences, Wuhan University, 430079 Wuhan, China
[b] GIS Technology, OTB, Delft University of Technology, Delft, The Netherlands
[c] Chinese Antarctic Center of Surveying and Mapping, Wuhan University, 430079 Wuhan, China

ARTICLE INFO

Article history:
Received 23 September 2013
Received in revised form 18 March 2014
Accepted 19 March 2014
Available online 13 April 2014

Keywords:
Data matching
Multi-scale databases
Contextual information
Relaxation labeling
Cartographic generalization

ABSTRACT

The aim of matching spatial data at different map scales is to find corresponding objects at different levels of detail (LODs) that represent the same real-world phenomena. This is a prerequisite for integrating, evaluating and updating spatial data collected and maintained at various scales. However, matching spatial data is not straightforward due to the ambiguities caused by problems like many-to-many correspondence, non-systematic displacement and different LODs between data sets. This paper proposes an approach to matching areal objects (e.g. buildings) based on relaxation labeling techniques widely applied in pattern recognition and computer vision. The underlying idea is to utilize contextual information (quantified by compatibility coefficient) in an iterative process, where the ambiguities are reduced until a consistent matching is achieved. This paper describes (1) a domain-specific extension to previous relaxation schemes and (2) a new compatibility coefficient that exploits relative relationships between areal object pairs in spatial data. Our approach were validated through extensive experiments using building data sets at 1:10k and 1:50k as an example. Our contextual approach showed superior performance against a non-contextual approach in general and especially in ambiguous situations. The proposed approach can also be applied to matching other areal features and/or for a different scale range.

1. Introduction

Data matching is important for many spatial data processes and applications. It aims at finding corresponding (or homogeneous) elements in heterogeneous data sources that represent the same real-world entities. Data matching is widely used in image registration, as well as for update, maintenance and integration of spatial data within and across data infrastructures (Ruiz et al., 2011). All these processes need data matching (i.e. finding correspondence relationships) to manage the presence of multiple representations of the same geographical area. This can be done in multiple representation databases (MRDBs) (Buttenfield and Delotto, 1989; Balley et al., 2004).

In addition to the syntactic and semantic differences, which are common for all kinds of data, spatial data add to the heterogeneity of multiple representations another dimension, i.e. the scale (level of detail or resolution) dimension. To ensure data consistency, linkages must be established between corresponding objects at different scales (Stoter et al., 2009a; Zhang et al., 2013), so that updates can be propagated to representations of the same objects at smaller scales (Kilpeläinen, 2000).

The correspondence relationships are also useful for assessing the quality of spatial data against a reference data source. In recent years, large amount of geographic content is voluntarily collected by citizens. To obtain more consistent or semantically richer data, such data should be firstly evaluated against a reference data source before it can be accepted as good data (Girres and Touya, 2010; Koukoletsos et al., 2012).

This paper focuses on the matching of features represented by polygons over different map scales. More specifically, we will match buildings in data sets ranging from large to mid-scales. Buildings are dominant areal features on topographic maps and with roads they require most updating in topographic data sets over all scale ranges (Stoter et al., 2009a,b). The matching of buildings of different scales is complex due to the ambiguities involved (see Section 1.1). Hence our approach uses contextual information to improve the performance. The approach can be easily adopted for other areal features.

* Corresponding author. Tel.: +86 13407118100.
E-mail addresses: xiang.zhang@whu.edu.cn (X. Zhang), tinghuaai@gmail.com (T. Ai), j.e.stoter@tudelft.nl (J. Stoter), xi.zhao@whu.edu.cn (X. Zhao).

http://dx.doi.org/10.1016/j.isprsjprs.2014.03.010
0924-2716/

1.1. Difficulties in matching multi-scale data

We assume that the multi-scale data has already been unified in a reference system so that they are roughly aligned. Consequently any object is within limited distance to the corresponding object. However, often more than one candidate is within limited distance which causes ambiguities and makes the matching not straightforward. This has several reasons.

First, many-to-many correspondence is an intrinsic component of multi-scale data. This is due to inherent changes in scale, at which real-world phenomena are represented. In cartography, generalization operations are used to generate spatial data at different map scales. For instance, the aggregation operation combines several objects into one object, creating a n-to-1 correspondence. The typification operation replaces objects at the larger scale by fewer objects at the smaller scale while keeping the initial spatial distribution (Fig. 1(a)). This leads to a n-to-m correspondence ($n > m$). In signal or image processing, 2D (and 3D) objects do not only combine at a coarse scale but also split (Lindeberg, 1994). This sometimes also occurs in cartography that an object can be split into more objects at a small scale.

Second, positional discrepancy between objects in two data sets may occur because of displacement. Displacement reduces visual clutter (specifically when symbolization causes objects to be larger on the map than they are in reality) and hence improves the readability of the map. Not all objects will be displaced and, even for the moved objects, the direction and amount of the movement can vary greatly. Therefore, the discrepancy between two data sets is non-systematic (see e.g. Fig. 9). This makes the matching difficult.

Third, objects at the smaller scale have been simplified. So matching based on the similarity of shapes is complicated. Furthermore, different generalization operations are usually applied together, which makes it difficult (even for the human visual system) to properly identify the corresponding objects (also see Fig. 9).

In ambiguous cases, matching depends largely on the context. In different contexts the same object may be matched differently, even by humans. For example, it is reasonable to match t_1 to s_2 and s_3 in Fig. 1(b). In the context of Fig. 1(c), t_1 is better matched to s_1 and t_2 to s_2 and s_3.

This paper proposes an approach that effectively incorporates contextual information to improve the matching in ambiguous situations.

1.2. Related work

Data matching has been studied in the geospatial domain for decades, originating in *map conflation*, a task to combine spatial data from different sources (Saalfeld, 1985, 1988); see Ruiz et al. (2011) for a review. Many recent studies focused on the development of either spatial or non-spatial measures that quantify the similarity between intended matching pairs. For instance, Beeri et al. (2005) developed a spatial join algorithm that matches objects using positions of their centroids. To match linear features, other information such as direction, shape and name are also used (Walter and Fritsch, 1999; Zhang and Meng, 2007; Raimond and Mustière, 2008). To match networks at different scales, Volz (2006) and Mustière and Devogele (2008) proposed to use topological structures of road intersections. On the other hand, Tong et al. (2009) and Zhang et al. (2012) proposed matching methods that effectively combine evidence from multiple measures. The latter used a learning-based framework, which avoids arbitrary weighting.

There are only a few approaches which incorporate contextual information. For example, Samal et al. (2004) described a method that measures the contextual similarity between buildings to be matched. In their work, landmarks are used as the context, which are connected to the target object in a star graph. The contextual similarity is calculated by the total displacement between two aligned graphs. This approach was extended by Kim et al. (2010) who used a triangulation instead to structure landmarks. They calculated the contextual similarity by areas and perimeters of the triangles around the target building. However, their contextual approaches have two main difficulties. First, landmarks themselves should be matched beforehand, which may not be easy, especially for multi-scale data. Second, the landmarks are sparsely distributed in the data. If objects are very close compared to their distance to the landmarks, it would be tricky to distinguish between them using their contextual measures.

In essence, the above two methods transform contextual information into descriptors, such that the similarity between contexts can be computed directly. Similar descriptors are also used in computer vision such as *SIFT* and *shape context* (Lowe, 1999; Belongie et al., 2002; Mikolajczyk and Schmid, 2005). When matching spatial objects at different scales, the spatial relationships between the target object and its surrounding objects at the same scale are also important. The influence of these relationships on determining the matching may vary in an iterative manner. Such contextual information is difficult to capture with a static descriptor, but can be incorporated by another type of approaches which are reviewed as follows.

A family of iterative algorithms introduced by Rosenfeld et al. (1976) has been extensively studied and became a standard technique in pattern recognition and computer vision (Pelillo, 1997). These are generally known as *relaxation labeling processes* and attempt to incorporate contextual information to arrive at a consistent result in labeling and matching problems where ambiguities may reduce the performance of non-contextual methods (Hummel and Zucker, 1983). Many probabilistic relaxation schemes were proposed for matching point and linear features (Christmas et al., 1995; Zheng and Doermann, 2006; Lee and Won, 2011). They differ in the definition of the compatibility coefficient and the rule used to update matching probabilities in the iterative process. These

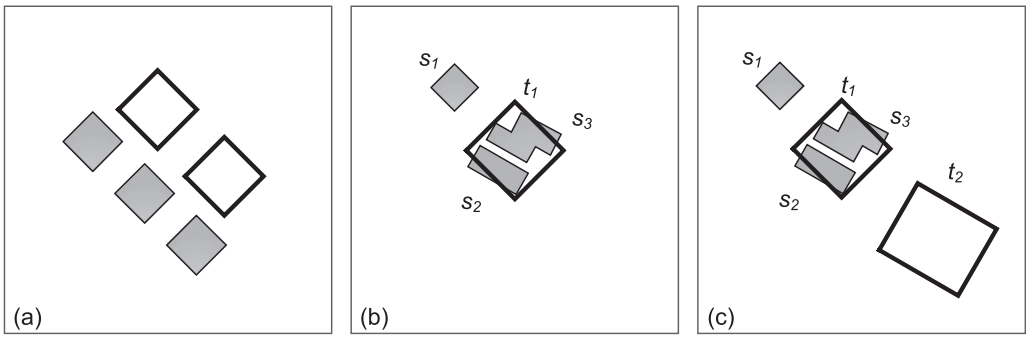


Fig. 1. Matching of multi-scale data exemplified (larger and smaller scale objects are in gray and white respectively).

techniques were recently applied in the geospatial domain to the matching of road networks (Song et al., 2011; Yang et al., 2013).

The basic assumption of the relaxation labeling is that compatibility relationships in the local neighborhood are retained under transformations, outliers, noises and occlusions. This is especially interesting for our case, where spatial relationships are preserved as much as possible for data at different scales. However, in this reviewed work the contextual information is mostly specific to point and linear features. On the contrary, in our research areal features need to be matched. In addition, matching in computer vision generally enforces one-to-one correspondence between points. Many-to-many correspondence, which is inherent in matching multi-scale spatial data (Section 1.1), should be explicitly handled.

This paper proposes a matching approach which extends existing relaxation labeling techniques in domain-specific ways to overcome the above-mentioned problems (Section 2). These include a domain-specific extension to standard relaxation schemes, which can handle many-to-many relationships well for our matching task, as well as a new compatibility coefficient for matching areal features in spatial data. For the compatibility, we exploit relative relationships (i.e. relative positions, orientations, sizes and shapes) between object pairs and propose ways to quantify them. Besides, several adaptations are made for matching areal features at multiple scales. The presented approach are tested and discussed through extensive experiments in Section 3. This paper concludes in Section 4.

2. Method

This section describes our method to match areas in the form of buildings (or groups of buildings) in data sets at different map scales. The matching methodology is the most important contribution of this paper and consists of several parts. An overview of each part is given below.

- First, the matching problem is formulated as a relaxation labeling process, where important formulas are presented and extended for matching data at different scales (Section 2.1). An algorithm of the process is also presented. This section describes the whole relaxation-based matching framework.
- Second, spatial context of an object is captured by a neighborhood structure, which is needed by a relaxation labeling process. We define neighborhood structures suitable for our case in Section 2.2.
- Third, the most important part of a relaxation labeling process is to formalize contextual information into a compatibility coefficient. Section 2.3 proposes a new compatibility coefficient which measures the similarity between relative relationships of object pairs.
- Finally, a workflow is designed to reduce the processing time by dividing the geographic area into reasonably small cells (Section 2.4).

2.1. A relaxation-based matching framework

This section formulates the matching problem as a relaxation labeling process. Let $T = \{t_1, t_2, \ldots, t_I\}$ and $S = \{s_1, s_2, \ldots, s_J\}$ be two sets of objects to be matched. In our case, T is the smaller scale data and S is the larger scale data, but this is not a requirement for our matching approach. The goal is to find every possible object pair (t_i, s_j) that either represents the same real-world object in a 1-to-1 case, or one is a part of the other object in a n-to-m case. Due to the inherent ambiguities in our matching task (Section 1.1), it is reasonable to assign each (t_i, s_j) with a matching probability $p_{ij} \in [0, 1]$. The probabilities of all object pairs can be represented by a matrix P with dimension $I \times J$. It is useful to denote the neighbors of t_i (i.e. objects that fall within some distance to t_i) as $t_h \in \mathcal{N}_i$ and those of s_j as $s_k \in \mathcal{N}_j$.

The basic idea of relaxation labeling is that it iteratively updates p_{ij} (and hence P) until a global consistency is achieved. This iterative process reduces local ambiguities by exploiting contextual information which is measured by a compatibility coefficient. The compatibility coefficient, $r_{ij}(h,k)$, quantifies the degree of agreement between t_i matched to s_j and t_h matched to s_k. As a key contribution, we propose to calculate the agreement by examining the relative relationships (e.g. relative position and orientation) between object pairs. If the relative relationships of an object pair (t_i, t_h) are similar to its corresponding pair (s_j, s_k), they are compatible and thus a high value of $r_{ij}(h,k)$ is obtained. If not, then they are incompatible. For example, the matching of (i,j) in Fig. 2 receives strong support from the matching of (h,k); whereas it receives less support from (h,k') because the relative relationships of (i,h) are not compatible (similar) to those of (j,k'). The calculation of $r_{ij}(h,k)$ is explained in detail in Section 2.3.

In the iterative relaxation process, (t_i, s_j) receives total support (q_{ij}) from its neighbors based on which p_{ij} is updated. A useful support function was proposed by Ranade and Rosenfeld (1980) and Hummel and Zucker (1983):

$$q_{ij}^{(t)} = \sum_{h \in \mathcal{N}_i} \max_{k \in \mathcal{N}_j} \left\{ r_{ij}(h,k) \cdot p_{hk}^{(t)} \right\}. \tag{1}$$

Note that $r_{ij}(h,k)$ is weighted by p_{hk}, because support from the neighbors t_h and s_k should also rely on their matching probability p_{hk}. A low value of p_{hk} is less likely to give strong support to the matching of (t_i, s_j). An alternative support function is given by Hummel and Zucker (1983):

$$q_{ij}^{(t)} = \sum_{h \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_j} r_{ij}(h,k) \cdot p_{hk}^{(t)}. \tag{2}$$

The above two support functions are termed *arithmetic average support function* (Christmas et al., 1995; Wang and Hancock, 2008). The main difference between the two is as follows. For each t_h there may be many $s_k \in \mathcal{N}_j$, in Eq. (1) only one s_k with the maximum support value is taken; while in Eq. (2) support is summed over $s_k \in \mathcal{N}_j$. For both functions, support is summed over $t_h \in \mathcal{N}_i$.

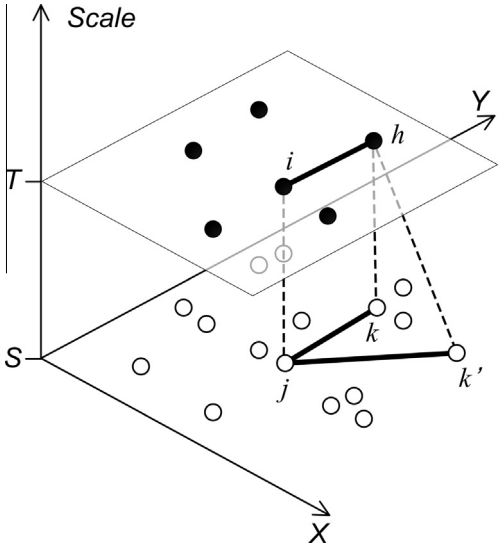


Fig. 2. Relaxation labeling demonstrated for matching multi-scale spatial data: the matching of i and j relies on their neighbors h and k (sets of objects T (•) and S (○) are represented in parallel *2-D* planes ($X-Y$); scale is represented as the third dimension).

Clearly, the two functions would yield q_{ij} of different magnitude (of which q_{ij} from Eq. (2) is larger). In our experiments we tested both functions in terms of accuracy (Section 3.2.1) and and efficiency (Section 3.2.6).

To update each p_{ij} in P, the *radial projection* update rule of Parent and Zucker (1989) proved to be useful, which enables a faster convergence to a consistent result (Pelillo, 1997). The original update rule is defined as:

$$p_{ij}^{(t+1)} = \frac{p_{ij}^{(t)} + q_{ij}^{(t)}}{1 + \sum_{j=1}^{J} q_{ij}^{(t)}}. \tag{3}$$

The above *radial projection* update rule is defined in a way such that the constraint, $\sum_{j=1}^{J} p_{ij} = 1$, is satisfied after the relaxation process (Parent and Zucker, 1989). This means that, under the assumption that we select final matches based on the criterion: $p_{ij} > 0.5$, only one s_j may be matched to t_i. This enforces one-to-one correspondence. Our experiments also confirmed this, as will be shown later on. Although one-to-one correspondence is commonly required in point matching in the computer vision domain (Chui and Rangarajan, 2003; Zheng and Doermann, 2006; Lee and Won, 2011), it is not suitable in our case where many-to-many correspondence has to be considered.

To allow for many-to-many correspondence, we modify the constraint to be satisfied as $\sum_{j=1}^{J} (p_{ij})^2 = 1$, such that for each t_i more than one s_j may have a matching probability higher than 0.5.[1] As another important contribution, we rewrite Eq. (3) in the following form to enforce the modified constraint:

$$p_{ij}^{(t+1)} = \frac{p_{ij}^{(t)} + q_{ij}^{(t)}}{1 + \sqrt{\sum_{j=1}^{J} \left(q_{ij}^{(t)}\right)^2}}. \tag{4}$$

The theoretical background of Eqs. (3) and (4) is given in Appendix A. The relaxation labeling process works as follows: if the matching of t_i and s_j is strongly supported by many of their neighbors t_h and s_k, their matching probability p_{ij} increases; p_{ij} decreases if they receive small support from their neighbors. Note that the change of p_{ij} is highly dynamic because p_{hk} in Eqs. (1) and (2) is also updated as such. So the value of p_{ij} may go back and forth until the consistency is propagated throughout the whole network (see e.g. Fig. 13).

Algorithm 1. A relaxation-based matching algorithm

```
Initialization of matching matrix P^(0);
repeat      ▷Update P^(t+1) for t : 0 → ∞
  for all t_i ∈ T and s_j ∈ S do
    for all t_h ∈ N_i and s_k ∈ N_j do
      Calculate r_ij(h,k);
      Read p_hk^(t) from P^(t);
    end for
    Calculate q_ij^(t) according to Eq. (1) or (2);
    Read p_ij^(t) from P^(t);
    Calculate p_ij^(t+1) according to Eq. (4);
  end for
  ΔP ← P^(t+1) − P^(t);
  Find Δp_max such that ∀Δp ∈ ΔP, |Δp_max| ⩾ |Δp|;
until |Δp_max| < Δp_stop
```

The relaxation-based matching procedure is sketched out in Algorithm 1. Several issues should be noted when applying this algorithm. First, we choose to use a Naive Bayes based matching approach (Zhang et al., 2012) to initialize the matching matrix $P^{(0)}$. This matching approach is a probabilistic non-contextual method, so we use it also as a comparison to the proposed contextual approach. The choice however seems to be not that important, as our experiments show that the new algorithm did not critically rely on the initialization method (Section 3.2.5).

Second, to calculate Eq. (4), q_{ij} is normally summed over $j \in [1, J]$ ($s_j \in S$) in order to update p_{ij}. This is practically inefficient, but it can be improved for specific problems. In our case, potential matching candidates of t_i (denoted as C_i) lie in some limited distance to t_i:

$$C_i = \{s_j : d(t_i, s_j) \leqslant d_\tau\}, \tag{5}$$

where $d(\cdot)$ is the Euclidean distance function; d_τ is a distance threshold. For those s_j that are not in C_i, we set their matching probabilities to zero. By localizing the matching candidates, one only need to sum q_{ij} over $s_j \in C_i$ in order to evaluate Eq. (4). This reduces the computational complexity of Eq. (4) to a constant level which does not change as T and S grow. Note that d_τ is problem specific, and in our case it can be estimated from data or can be known if specifications (e.g. the maximum allowed displacement) of data are accessible.

The third issue is when to stop the iteration (the outmost loop) in Algorithm 1. Both theoretical and practical work shows that the relaxation labeling process is guaranteed to converge to a consistent solution (Hummel and Zucker, 1983; Pelillo, 1997; Zheng and Doermann, 2006). This means that the matching probability matrix P gradually becomes stable during the iteration process. By comparing values in P at consecutive iterations, $|\Delta p_{max}|$ becomes smaller as the iteration goes on. If $|\Delta p_{max}|$ is smaller than the threshold Δp_{stop}, the iteration stops (in our experiments we set Δp_{stop} to 0.0005). Our experiments suggest that this criterion can be relaxed in some circumstances (Section 3.2.6).

Besides, the neighborhood $(\mathcal{N}_i, \mathcal{N}_j)$ and compatibility coefficient $(r_{ij}(h,k))$ are important aspects in the algorithm and are not clearly defined here. We will define these aspects in Sections 2.2 and 2.3, respectively.

2.2. Defining neighborhood structures

There is no single definition of neighborhood for a set of objects that is useful for all matching tasks. For instance, one can define the neighbors of t_i as all other objects in T. Such a neighborhood structure definitely works but it considerably increases the processing time. In practice, it would be useful to distinguish between the neighborhood structure of local and usually continuous influence on other objects and a thresholding of the influence. A local structure is usually assumed in computer vision because after transformation the local neighborhood of an object is generally preserved, but the relationships between objects that are far apart may be lost (Zheng and Doermann, 2006; Lee and Won, 2011). The same is true for multi-scale spatial data. In Lee and Won (2011), the neighborhood of each shape point is defined by a fixed radius. This definition is only applicable where the shapes to be matched are represented by uniformly spaced points. If objects are irregularly distributed (as in our case), a fix radius sometimes fails to capture any neighbor. In such cases p_{ij} will not change in the iterative updating process as no support information is available (i.e. $q_{ij} = 0$).

A more adaptive definition is based on the K-nearest neighbors, which is translation, rotation and scale invariant (Zheng and Doermann, 2006; Sidibe et al., 2007). However, it is still tricky to decide

[1] This is a domain-specific rather than a generally applicable constraint (see Appendix A for details).

the value of K, and the K-nearest neighbors may be clustered in some directions of the target object, thus missing important neighbors in some other directions.

In this paper, we use Delaunay triangulation (DT) to define the neighborhood of objects. This approach has the same merits as with the K-nearest approach, since the nearest neighbor graph is a subgraph of DT. It always ensures that there are enough neighbors around the target object regardless of data distribution. In addition, it provides a more powerful structure where the neighbors are well distributed in all directions of the target object. Another difference lies that in Zheng and Doermann's approach a topological neighborhood is defined, meaning that the influence of an object is either zero (if it is in the neighborhood) or one (if not). Our approach models continuous influence of neighboring objects.

Formally, DT can be represented by a graph $G_{DT} = (V, E)$, where V is a set of vertices (in our case, objects in T or S) and $E \subset V \times V$ is a set of edges which capture the proximity relationship between vertices. In our case, we use DT to define the neighborhood for $t_i \in T$ only (i.e. the smaller scale data). More specifically, first-order neighbors (i.e. objects directly connected by edges of DT) are used (see e.g. Fig. 3):

$$\mathcal{N}_i = \{t_h : (t_i, t_h) \in E_T\}, \tag{6}$$

where E_T denotes the edges of the DT that takes T as input. Because the larger scale (i.e. more detailed) data set S contains many more objects than T ($I < J$), using the same neighborhood definition for s_j is not appropriate. For example, if DT is used to identify neighbors of j in Fig. 3, an important object k, which together with h gives maximum support to (i,j), will not be included. Likewise, many other important objects will not be in the neighbors of j. This definitely reduces the performance of the relaxation labeling. A similar problem is reported in Zheng and Doermann (2006), where under a large outlier ratio (i.e. extra points being added to the transformed data) their K-nearest neighbor based relaxation labeling failed to give satisfactory results.

To overcome this problem, we define the neighbors of s_j as the objects closely related to t_h such that all important neighbors are considered:

$$\mathcal{N}_j = \bigcup_{h \in \mathcal{N}_i} \{s_k : d(t_h, s_k) \leqslant d_\tau\}, \tag{7}$$

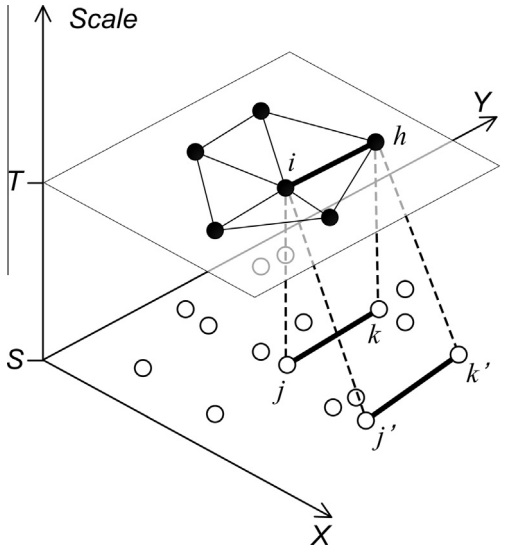


Fig. 3. Neighborhood of i defined on top of Delaunay triangulation; both (j', k') and (j, k) have similar relative (binary) relationships to (i, h), so $r_{ij}(h, k)$ and $r_{ij'}(h, k')$ would be more or less the same if unary measures are not incorporated.

where d_τ is the same as with (5). Because there are many $t_h \in \mathcal{N}_i$, a union of the sets related to all t_h is necessary. Note that some s_k that is close to s_j but far from t_h may not be considered s_j's neighbors in this definition. This is nevertheless not a problem, because we set $p_{hk} = 0$ if $d(t_h, s_k) > d_\tau$. So by omitting s_k with $p_{hk} = 0$, one does not change the results of Eqs. (1) and (2), and therefore, the relaxation process remains the same.

To summarize, two different definitions of neighborhood are given to sets T and S. The former is defined on top of Delaunay triangulation to handle the irregular data distribution. The latter is defined in relation to the former one to address the problem due to the matching of multi-scale data.

2.3. Quantifying contextual information

This section explains the calculation of compatibility coefficient $r_{ij}(h, k)$, by which contextual information is quantified in our approach. To calculate $r_{ij}(h, k)$, we exploit the similarity of relative (binary) relationships between object pairs. This is because relative relationships in the larger scale data are generally kept in the next smaller scale data (Steiniger and Weibel, 2007). More specifically, relative positions, orientations, sizes and shapes are measured and compared to calculate the correlation between an areal feature pair (t_i, t_h) and its corresponding pairs (s_j, s_k).

However, we found that the use of binary relationships alone cannot always identify correct matches. For example, j and j' in Fig. 3 may be matched to i with similar probabilities no matter how far j' is located. This is because both (j', k') and (j, k) have similar relative relationships to (i, h). The problem here is that absolute (unary) measures that quantify the similarity between t_i and its matching candidate s_j are overlooked during the iterative update process. Such unary measures were also missing in most previous relaxation labeling based matching approaches (Christmas et al., 1995). In this paper, unary and binary measures are used together to calculate $r_{ij}(h, k)$, such that distant and dissimilar matching candidates s_j are considered less likely to be a corresponding object of t_i.

In the following, we firstly decompose the compatibility coefficient into components of position, orientation, size and shape, where both unary and binary measures are given (Sections 2.3.1, 2.3.2, 2.3.3 and 2.3.4). The relative position component is adapted from Lee and Won (2011), while others are proposed to improve the matching of areal features. Then these components are combined to define the total compatibility coefficient in Section 2.3.5.

2.3.1. Relative positions

Relative positions are measured for two vectors (t_i, t_h) and (s_j, s_k) and their similarity is further divided into two sub-components (Fig. 4). They are the relative distance (*RelDis*) and relative direction (*RelDir*) which are defined as follows:

$$RelDis(ih; jk) = 1 - |d(i,h) - d(j,k)| / \max_{m \in \mathcal{N}_i, n \in \mathcal{N}_j} \{d(i,m), d(j,n)\}, \tag{8}$$

$$RelDir(ih; jk) = 1 - \Delta\alpha(ih; jk) / \Delta\alpha_{max}, \tag{9}$$

where $\Delta\alpha(ih; jk) = |\alpha(i,h) - \alpha(j,k)|$ is the directional deviation ($\Delta\alpha$ for short). For two vectors $\Delta\alpha$ is in the range $[0, \pi]$, so normally one can use $\Delta\alpha_{max} = \pi$ to normalize *RelDir*. In our matching case, however, two vectors are not compatible if their relative direction is larger than $\pi/2$. To enforce this constraint, we trim $\Delta\alpha$ to the range $[0, \pi/2]$ (i.e. $\Delta\alpha = \pi/2, \text{if} \Delta\alpha > \pi/2$). Accordingly, we set $\Delta\alpha_{max} = \pi/2$ in our approach.

RelDis and *RelDir* range from 0 to 1. High values indicate that the relative distance and direction of (t_i, t_h) are similar to those of (s_j, s_k). This similarity is weighted by the distance between t_i and t_h (w in Fig. 4):

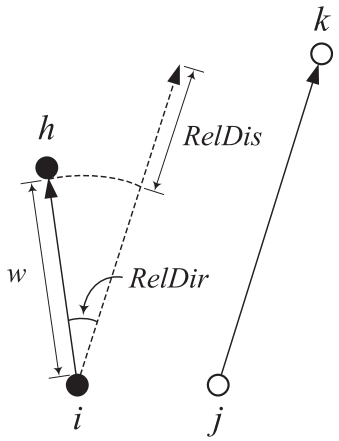


Fig. 4. Two sub-components: relative distance (*RelDis*) and relative direction (*RelDir*) measure the similarity between relative positions of (i,h) and (j,k) (after Lee and Won (2011)).

$$w(i,h) = 1 - d(i,h)/\max_{m\in N_i}\{d(i,m)\}. \tag{10}$$

The total similarity between relative positions is defined by multiplying from Eqs. (8)–(10):

$$Rel_1(ih;jk) = RelDis(ih;jk)\cdot RelDir(ih;jk)\cdot w(i,h). \tag{11}$$

Note that Rel_1 also ranges from 0 to 1. A high value means that (s_j,s_k) is very similar to (t_i,t_h) in terms of relative positions; a low value means that they are dissimilar. Besides the relative positions, the similarity between absolute positions of t_i and s_j is also important:

$$Abs_1(i,j) = 1 - d(i,j)/d_\tau, \tag{12}$$

where d_τ is a distance threshold used as a normalizing factor. This d_τ is the same parameter as in definition (5). Because we only consider $s_j \in C_i$ (see definition (5)), the condition $d(i,j) \leqslant d_\tau$ always holds in Eq. (12).

2.3.2. Relative orientations

Orientation is a discriminative characteristic of building features. Many corresponding pairs can hardly be identified without comparing the relative orientations in the local neighborhood. To quantify the orientation of buildings, we found the *wall statistical weighting* measure (Duchêne et al., 2003) useful as wall orientations are well kept in our multi-scale data (e.g. Fig. 7). In this measure, each direction comes with a confidence indicator, measuring how many walls (roughly) parallel or perpendicular to this direction. Among all directions, one can choose major directions as final orientations.

To better handle relative orientations, we use an adaption of *wall statistical weighting* described in Zhang et al. (2012) to distinguish buildings with significant orientations (e.g. elongated rectangles) from those without significant orientations (e.g. squares). The adapted orientation β is in the range $[0,\pi)$.

The relative orientation between t_i and t_h is defined as: $\theta_{ih} = \beta_h - \beta_i$ ($\theta_{ih} \in [-\pi,\pi)$). As orientation β can also be represented by $\beta \pm \pi$, the maximum relative orientation is $\pi/2$ (i.e. $\theta_{ih} \in [-\pi/2,\pi/2)$). This is for normal cases where both objects have significant orientations. If one of the objects has no significant orientation (e.g. squares or circles), the orientation difference is limited to $\pi/4$ (i.e. $\theta_{ih} \in [-\pi/4,\pi/4)$). Likewise, θ_{jk} can be defined in the same way. Relative orientations between buildings are illustrated in Fig. 5(a).

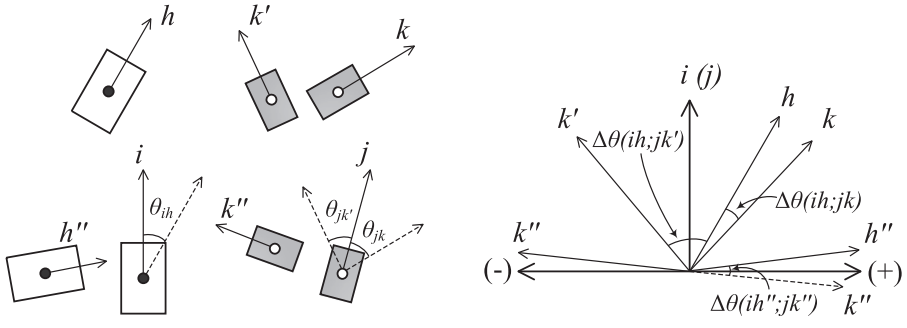
Note that both θ_{ih} and θ_{jk} are signed deviations. That is, although building orientations may have the same absolute deviation, they may have different directions. This is vital when two relative orientations are compared. To demonstrate this, we project β_i and β_j onto the same vertical axes in Fig. 5(b), then orientations of other buildings, i.e. h and k, can be mapped relative to i and j. Orientations to the right side of $i(j)$ have positive θ regarding to $i(j)$; whereas those to the left side have negative θ. It is clear in Fig. 5(a) that $\theta_{jk'}$ and θ_{jk} differ greatly as they are in different direction. We define similarity between relative orientations as follows:

$$Rel_2(ih;jk) = 1 - \Delta\theta(ih;jk)/\Delta\theta_{max}, \tag{13}$$

where $\Delta\theta(ih;jk) = |\theta_{ih} - \theta_{jk}|$. Fig. 5(b) confirms that the relative orientation of (j,k) is similar to that of (i,h) but is dissimilar to that of (j,k').

As both θ_{ih} and θ_{jk} are in $[-\pi/2,\pi/2)$, it follows that $\Delta\theta(ih;jk) \in [0,\pi)$. However, a deeper analysis shows that the maximum value of $\Delta\theta(ih;jk)$ should not exceed $\pi/2$. For instance, the calculated angular deviation between (i,h'') and (j,k'') in Fig. 5 is larger than $\pi/2$. But because $\beta_{k''}$ can also be represented by $\beta_{k''} \pm \pi$ (see the dashed arrow in Fig. 5(b)), the actual angular deviation $\Delta\theta(ih'';jk'')$ is the smaller angle. This is consistent with our observation that (i,h'') and (j,k'') in Fig. 5(a) have similar relative orientations. That is, both h'' and k'' are roughly perpendicular to i and j, respectively. Because of this, we fit $\Delta\theta(ih;jk)$ in $[0,\pi/2]$ (i.e. $\Delta\theta = \pi - \Delta\theta$, if $\Delta\theta > \pi/2$), and set $\Delta\theta_{max} = \pi/2$ in Eq. (13).

Again, if one of the buildings in $\{i,j,h,k\}$ does not have a significant orientation, $\Delta\theta(ih;jk)$ is limited to $[0,\pi/4]$. Therefore, we apply $\Delta\theta = \pi/2 - \Delta\theta$, if $\Delta\theta > \pi/4$. Such a treatment is necessary as some elongated buildings at larger scales may be represented by less elongated rectangles or squares at smaller scales. Their orientations are actually the same. Without this treatment the computation of $\Delta\theta(ih;jk)$ may give a totally different result, which would reduce the matching performance.



(a) Orientations and relative orientations (b) Orientations in a projected space

Fig. 5. Demonstration of relative orientations and their deviations.

Rel_2 in Eq. (13) ranges from 0 to 1. A high value indicates that the relative orientation between s_j and s_k is highly correlated to that between t_i and t_h; vice versa. Besides the relative orientations, the similarity between absolute orientations of t_i and s_j is also useful:

$$Abs_2(i,j) = 1 - \frac{\theta_{ij}}{\pi/2}, \tag{14}$$

where $\theta_{ij} = |\beta_i - \beta_j|$ is similar to the definition of θ_{ih} but it is an absolute deviation. So θ_{ij} is in the range $[0, \pi/2]$ and, if one of the buildings does not have a significant orientation, θ_{ij} is limited to $[0, \pi/4]$.

2.3.3. Relative sizes

This component compares if the relative size of (s_j, s_k) is correlated to that of (t_i, t_h). The relative size relationship is defined as the ratio of sizes:

$$\begin{cases} \phi_{ih} = size(i)/size(h) \\ \phi_{jk} = size(j)/size(k) \\ \phi_{ih} = 1/\phi_{ih}, \phi_{jk} = 1/\phi_{jk}, \text{ if } \phi_{ih} > 1 \end{cases}. \tag{15}$$

The relative size can be qualitatively characterized by 'similar', 'smaller' or 'bigger'. Note that in the above formula, we always keeps ϕ_{ih} in the range $[0, 1]$. If $\phi_{ih} > 1$, both ϕ_{ih} and ϕ_{jk} are inverted regardless of the value of ϕ_{jk}. The deviation between relative sizes is defined as $\Delta\phi(ih;jk) = |\phi_{ih} - \phi_{jk}|$. In addition to keeping the order of the relative sizes, the treatment in Eq. (15) has the following merit. If the relative sizes of (t_i, t_h) and (s_j, s_k) are qualitatively similar (e.g. either 'similar', 'bigger' or 'smaller'), $\Delta\phi(ih;jk)$ is relatively small; if for instance the relation of one pair is 'smaller' and that of the other is 'bigger', the deviation is relatively large.

For example, size ratios of the object pairs $(i,h), (j,k)$ and (j,k') in Fig. 6 are 2, 2.5 and 0.5, respectively. Since the ratio of (i,h) is larger than one, we calculate the inverse of these ratios. The result is $\phi_{ih} = 0.5, \phi_{jk} = 0.4$ and $\phi_{jk'} = 2$. It then follows that $\Delta\phi(ih;jk) = 0.1$ and $\Delta\phi(ih;jk') = 1.5$. This is consistent with our observation that (j,k) is very similar to (i,h) in terms of relative sizes, while (j,k') is rather different from (i,h). It can be further proven that $\Delta\phi(ih;jk)$ is always smaller than one if the relative sizes of (t_i, t_h) and (s_j, s_k) are qualitatively similar.

Because relative sizes are defined by ratios, $\Delta\phi(ih;jk)$ is unbounded. Hence we use the function in Ranade and Rosenfeld (1980) to convert the deviation to the similarity measure, which we found empirically to give good results:

$$Rel_3(ih;jk) = 1/(1 + (\Delta\phi(ih;jk))^2). \tag{16}$$

Rel_3 ranges from 0 to 1. A high value indicates a high degree of similarity between relative sizes. In the above examples, $Rel_3(ih;jk) \approx 0.99$ and $Rel_3(ih;jk') \approx 0.31$, meaning that (j,k) is more than (j,k') compatible to (i,h) in terms of relative sizes.

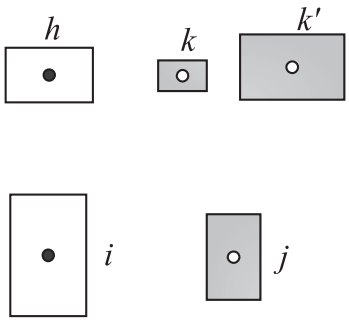


Fig. 6. Relative sizes demonstrated.

The similarity between absolute sizes of t_i and s_j is defined as a ratio. A thresholding is applied here to enlarge the sizes of objects at the larger scale to the threshold A_τ, which is scale dependent (see Section 3.2.3):

$$\begin{cases} size(j) = A_\tau, \text{if } size(j) < A_\tau \\ Abs_3(i,j) = \phi_{ij} = size(i)/size(j). \\ Abs_3(i,j) = 1/\phi_{ij}, \text{ if } \phi_{ij} > 1 \end{cases} \tag{17}$$

This is because during generalization, small objects at the larger scale are enlarged to make the smaller scale map more readable. This threshold effect is also described in Bard (2004) and Mackaness and Ruas (2007). For a small object s_j, its size can be much smaller than its correspondence t_i. A direct comparison between sizes would prevent the identification of correct corresponding objects. The treatment in Eq. (17) avoids such a problem (see Section 3.2.3).

2.3.4. Relative shapes

In this paper, we use elongation to quantify object shapes, as relative elongations are largely preserved for multi-scale data. Other measurements (e.g. concavity, squareness, etc.) may also be useful, but this requires further testing. We define the elongation of t_i as $elg(i) = Length/Width$, where $Width$ and $Length$ are measured from the rotated minimum bounding box of t_i. The relative elongations are defined in a similar way as relative sizes:

$$\begin{cases} \lambda_{ih} = elg(i)/elg(h) \\ \lambda_{jk} = elg(j)/elg(k) \\ \lambda_{ih} = 1/\lambda_{ih}, \lambda_{jk} = 1/\lambda_{jk}, \text{ if } \lambda_{ih} > 1 \end{cases} \tag{18}$$

The similarity between relative elongations are given by:

$$Rel_4(ih;jk) = 1/(1 + (\Delta\lambda(ih;jk))^2), \tag{19}$$

where $\Delta\lambda(ih;jk) = |\lambda_{ih} - \lambda_{jk}|$. Similar to relative sizes, relative elongations are designed in a way such that if the relative elongation of (t_i, t_h) is qualitatively similar to that of (s_j, s_k) (e.g. t_i is more elongated than t_h and s_j is more elongated than s_k), $\Delta\lambda(ih;jk)$ is small and $Rel_4(ih;jk)$ approaches to 1. If they are dissimilar (e.g. t_i is more elongated than t_h but s_j is less elongated than s_k), $\Delta\lambda(ih;jk)$ is large and $Rel_4(ih;jk)$ approaches to 0.

Due to the enlargement of small buildings or of their small parts (e.g. short edges), narrow objects s_j in the larger scale are usually more elongated than their corresponding t_i in the smaller scale. However, the change of elongation values is much more complex to model than the change of sizes (which is a simple threshold effect). Therefore, we will not use the similarity between absolute elongations of t_i and s_j in our matching approach.

2.3.5. Total compatibility coefficient

The total compatibility coefficient should be combined from the above relative and absolute measures, as each measure may contribute to the matching in some situations. The absolute (unary) measures are used to give more weight to those more similar pairs (t_i, s_j) in the iterative update process. The total compatibility coefficient $r_{ij}(h,k)$ is defined as the product of all binary and unary measures:

$$r_{ij}(h,k) = \prod_{u=1}^{4} Rel_u(ih;jk) \prod_{u=1}^{3} Abs_u(i,j), \tag{20}$$

where Rel_u is the binary measures defined in Eqs. (11), (13), (16) and (19); Abs_u is the unary measures defined in Eqs. (12), (14) and (17). Because both Abs_u and Rel_u are in the range $[0, 1]$, it follows that $r_{ij}(h,k) \in [0, 1]$. A high value of $r_{ij}(h,k)$ indicates a high degree of compatibility, and vice versa. The coefficient is used in the iterative

relaxation labeling process (Algorithm 1) until a consistent result is reached.

Here we explain how the relative relationships help arrive at a consistent matching in Fig. 1(c). Suppose that both t_1 and t_2 can be matched to s_1, s_2 and s_3 in an initial configuration, where t_1 seems to be more probably matched to s_2 and s_3. Let us examine the matching of (t_1, s_2) using contextual information. If t_1's neighbor t_2 was to be matched with s_3, the relative positions and orientations of (t_1, t_2) and (s_2, s_3) are not compatible (e.g. the relative direction between (t_1, t_2) and (s_2, s_3) is larger than $\pi/2$). So the matching of (t_1, s_2) is not supported by (t_2, s_3), and, in a similar way, not supported by (t_2, s_1). Likewise, the matching of (t_2, s_3) is not supported by its neighboring pairs. The only compatible configuration is that t_1 matches to s_1 and t_2 matches to s_2 or s_3. In this configuration the relative position, orientation, size and elongation of (t_1, t_2) are similar (compatible) to those of (s_1, s_2) or (s_1, s_3). In the iterative updating process, the matching probability of (t_1, s_1) increases while that of (t_1, s_2) and (t_1, s_3) decreases; for t_2, the matching probability of (t_2, s_2) and (t_2, s_3) increases while that of (t_2, s_1) decreases.

2.4. A workflow for matching areal features

To apply the proposed matching approach (Sections 2.1,) in practice more efficiently, a subdivision scheme that reduces the computation of Algorithm 1 to a manageable size is necessary. Here the buildings to be matched are subdivided into partitions by road network, so that each run the algorithm deals only with the buildings in one street block. Usually, buildings within a block have been generalized independently with respect to other blocks. Further, buildings may have been displaced in a different direction than those on the other side of the road, in which case the relative positions between these objects are not preserved. These reasons justify our use of road network as partitioning features.

The overall workflow of processing is to (1) partition building sets T and S into subsets using road network; (2) apply Algorithm 1 to match buildings within each partitioning unit; (3) stop if all the partitioning units are processed; and (4) do whatever postprocessing required and then select final matching pairs (t_i, s_j). Currently, the selection is based on the criteria $p_{ij} > 0.5$.

3. Experiments, results and discussion

3.1. Experimental design

The proposed matching approach was implemented in C++. In the following, we describe the data used to test our approach and the evaluation methods and criteria.

3.1.1. Test data

Two topographic data sets at 1:10k and 1:50k (from Kadaster, the Netherlands) were used to test our matching approach (Fig. 7), where the latter is generalized from the former. The geographic characteristics of the data are rich, covering urban, residential, industrial (commercial), suburban and rural areas. The generalization carried out for different data characteristics varies. For densely populated urban and residential areas, buildings are transformed into built-up areas. Since the matching of built-up areas and their corresponding building footprints is straightforward, we exclude those built-up areas from our experiments. After removing the buildings in built-up areas, the 1:10k and 1:50k data contain 2646 and 1637 buildings respectively.

Among the remaining buildings, those in the suburbs are expected to be the most difficult to match. Because there is not enough room to enlarge and displace small buildings, typification is used together with enlargement and displacement in suburban areas, which leads to ambiguities in the matching (Fig. 7(c)). Buildings in industrial areas, on the other hand, should be easier to

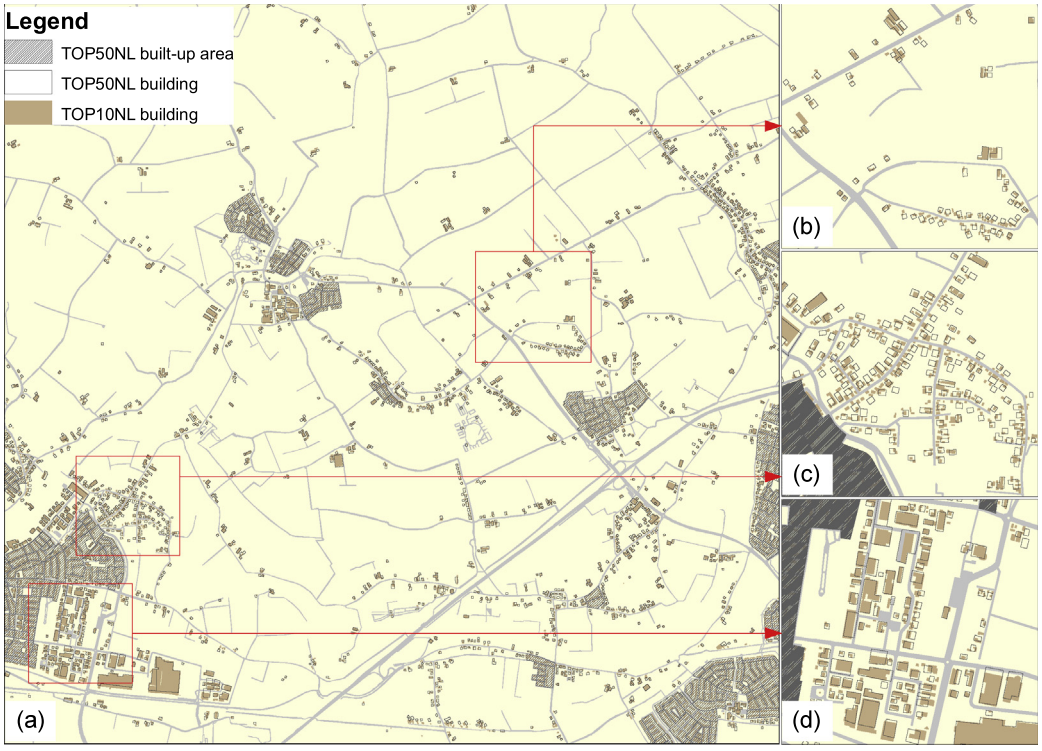


Fig. 7. Test data sets where 1:10k and 1:50k buildings are superimposed (a); samples of typical characteristics: (b) rural, (c) suburban and (d) industrial areas (Source: Kadaster, NL).

match, because most smaller scale buildings stay where they are in the larger scale data (Fig. 7(d)).

3.1.2. Evaluation methods

To evaluate the performance of the proposed matching approach, we asked cartographic experts to manually match our test data. The manual matching result is used as the 'ground truth', against which automated approaches can be evaluated. Matches found by both the algorithm and humans are true positives (*tp*); matches found by the algorithm but not by humans are false positives (*fp*); matches found by humans but not by the algorithm are false negatives (*fn*). The matching accuracy is measured by *precision*, *recall* and *Cohen's kappa coefficient* (κ). *Precision*=*tp*/(*tp*+*fp*) describes the rate that a pair matched by the algorithm was also found by humans. *Recall*=*tp*/(*tp*+*fn*) describes the rate that a pair matched by humans is also found by the algorithm. Cohen's kappa measures the agreement between the algorithmic and human matching results, taking into account the chance agreement.

Note that in highly ambiguous situations (e.g. suburban areas) different experts disagreed with each other. For those situations we combined their results such that more matching pairs were added as 'true' matches. In this way, we make sure that the real true matches are always included though the *recall* in such areas will be reduced. Another idea would be to define a minimum (i.e. all experts agreed) and a maximum (i.e. matched by at least one expert) set and to evaluate the approach for both sets. In this paper, we used the maximum set because in the manual matching process we did not record expert specific information.

As a comparison, we also implemented a non-contextual approach – the Naive Bayes (NB) based learning approach. The probabilistic model of NB was trained using samples from the manual matching result, which was then applied to match the test data (for details see Zhang et al., 2012). The matching result was also used to initialize the matching probability matrix for the subsequent relaxation process.

To get more insight, different versions of the proposed approach were tested. These versions differ in the use of compatibility and support function. The compatibility $r_{ij}(h,k)$ defined in Eq. (20) can be configured by using only some of its components (e.g. position, orientation, size and elongation). These versions of relaxation labeling are abbreviated in Table 1 for clarity.

We start from a version (LW_{01}^{+}) which extends the compatibility proposed in Lee and Won (2011) that uses relative positions only. However, we found that their original compatibility led to poor results in our case, so we start from LW_{01}^{+} where the similarity between absolute positions (Eq. (12)) is also considered. The use of unary measures together with binary relationships are justified in Section 2.3. The following versions RL_{01}, RL_{02} and RL_{03} in Table 1 are configured such that each version adds one more component to the previous version. The default support function used in these versions is Eq. (1), expect for those mentioned in Table 1. Versions RL_{03} and RL_{13} use exactly the same compatibility as defined in Eq. (20).

Further, we used roads at 1:50k to partition the buildings into 105 street blocks. The proposed matching algorithm and its various versions were applied sequentially for these street blocks. The experimental results are reported and discussed in Section 3.2.

Table 1
Abbreviation of the matching methods tested.

Abbrv.	Method description
NB	Naive Bayes based matching
	Below are the relaxation based matching schemes with different compatibilities and support functions
LW_{01}^{+}	Extended from Lee & Won's compatibility: {position (rel + abs)}
LW_{11}^{+}	LW_{01}^{+} with support Eq. (2)
RL_{01}	LW_{01}^{+} + {orientation (rel + abs)}
RL_{11}	RL_{01} with support Eq. (2)
RL_{02}	RL_{01} + {size (rel + abs)}
RL_{12}	RL_{02} with support Eq. (2)
RL_{03}	RL_{02} + {relative elongation}
RL_{13}	RL_{03} with support Eq. (2)

3.2. Results and discussion

3.2.1. General performance

We applied different versions of our approach as well as the NB-based approach to the test data. For the relaxation based approaches, we used $A_\tau = 400\ \text{m}^2$ and $d_\tau = 50$ m. The value of A_τ is suggested by the specifications of 1:50k data, so it is scale dependent. The value of d_τ was empirically determined. Their performance is shown in Table 2.

Clearly, all relaxation-based contextual approach outperformed the non-contextual approach (NB). Although NB obtained relatively high *precision*, its *recall* is relatively too low, which limits its overall performance (a relatively low *kappa* value). For instance, while many (1599) matches found by NB were also linked by human experts, there are still many (409) matches found by human experts that were missed by NB. This is also evident in Fig. 9(a), where many correct matches were not found by NB. All the relaxation-based methods found far more true positives (1830+) than NB while at the same time suppressing the number of false positives and false negatives.

In addition, Table 2 clearly shows that, by adding more components (i.e. relative and absolute measures) to $r_{ij}(h,k)$, the proposed matching approach gradually improved its overall performance (κ). This implies that different relative relationships contribute to our matching in certain situations. For example, situations highlighted in Fig. 9(b) cannot be found without the use of relative positions; those highlighted in Fig. 9(c) cannot be found without the use of relative orientations. The combination of different pieces of contextual information leads to an overall improvement in performance. But we also note that RL_{03} was only slightly better than RL_{02}. This means that the addition of relative elongations was not as informative as other relative relationships for the test data. Besides, the versions with the support function Eq. (2) performed slightly better than their counterparts with Eq. (1). This is because Eq. (2) exploits more contextual information (i.e. more $s_k \in \mathcal{N}_j$ are involved), though the improvement obtained was limited.

Finally, LW_{01}^{+} and LW_{11}^{+} only use positional information and ignore the information given by orientation, size and shape that is useful to areal features. Their matching results were not comparable to the other versions of our approach (e.g. correct matches highlighted in Fig. 9(c) was not found in Fig. 9(b)). Therefore, the versions from RL_{01} to RL_{13} would be more appropriate for matching areal features. The choice depends on how the relative relationships are preserved between the data (an issue specifically related to multi-scale data).

Table 2
Performance of different matching methods (for relaxation-based methods, $d_\tau = 50$ m was used).

Method	*tp*	*fp*	*fn*	*Prec.* (%)	*Recall* (%)	κ
NB	1599	169	409	90.4	79.6	0.809
LW_{01}^{+}	1852	245	156	88.3	92.2	0.871
LW_{11}^{+}	1865	234	143	88.9	92.9	0.879
RL_{01}	1863	212	145	89.8	92.8	0.885
RL_{11}	1863	210	145	89.9	92.8	0.885
RL_{02}	1833	151	175	92.4	91.3	0.893
RL_{12}	1840	156	168	92.2	91.6	0.894
RL_{03}	1832	146	176	92.6	91.2	0.894
RL_{13}	1837	147	171	92.6	91.5	0.896

3.2.2. Performance in different data characteristics

As discussed in Section 3.1.1, data matching may perform differently in different geographic characteristics. This is because the ambiguities involved vary greatly in different areas. This section takes a close look at this issue. To evaluate the matching performance in different areas, we manually classified the 105 street blocks into industrial, suburban and rural areas.

One can imagine that high accuracy can be obtained if the data is in general easy to match. As an example, we show the performance of RL_{02} compared with NB in Fig. 8, where the statistics (mean and standard deviation) were calculated from the 105 street blocks by category. Obviously, both the contextual and non-contextual methods obtained rather high accuracy in industrial areas. This is because the buildings in these areas are easy to match, i.e., more 1-to-1 relationships occur and most buildings have not been moved from their initial locations (see also Fig. 10). Even in such areas, a discernable rise in performance can be observed for the relaxation-based method (Fig. 8).

In ambiguous situations, superior methods should be able to further improve the matching. In suburban areas, the NB-based method performed poorly (its *recall* is extremely low). This is because the matching in such areas depends critically on contextual information. Hence the relaxation-based contextual methods performed considerably better than NB (Fig. 8), which is also clearly demonstrated in Fig. 9. We also noticed that the *recall* of RL_{02} is not very high in suburban areas, though its matching result shown in Fig. 9(c) is satisfactory. This is probably because, as we discussed in Section 3.1.2, in such areas more matches were added into the 'ground truth' as the experts disagreed with each other, which reduces the *recall* thereof in general. In rural areas, where the ambiguities are higher than in industrial areas but not as high as in the suburbs, the contextual methods (e.g. RL_{02}) also made a noticeable improvement against the non-contextual method (NB).

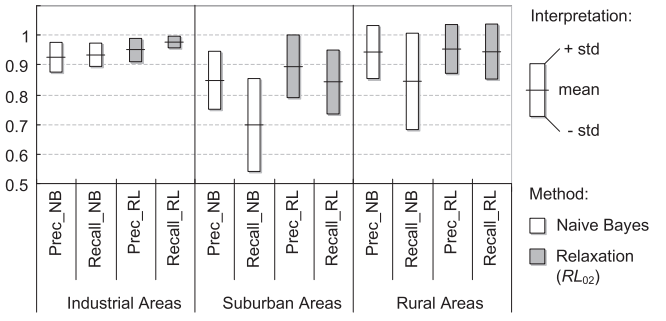


Fig. 8. Precision and recall obtained by Naive Bayes (NB) and our relaxation labeling (RL) based matching for different data characteristics (RL_{02} with $d_\tau = 50$ m was used).

3.2.3. Parameters of the algorithm

In general, the relaxation labeling framework is parameter free (Zheng and Doermann, 2006). But in our approach, we use a parameter d_τ to constrain the number of matching candidates and to define the neighborhood structure (see Definitions (5) and (7)). Ideally, the value of d_τ can be known for specific data, but it was determined empirically in our experiments to show the potential of the proposed approach.

Table 3 shows the performance of RL_{01} with varying values of d_τ. When d_τ is too small, some correct matches are not considered as matching candidates (e.g. low *tp* and high *fn* were produced with $d_\tau = 30$ m), which largely reduces the *recall* of the matching. As d_τ increases, the *recall* rises sharply and then drops smoothly, but the *precision* drops a bit more quickly. As a result, the overall performance (κ) goes up and down as d_τ increases. Still, the overall performance was better than the non-contextual method NB (see Table 2). The other versions higher than RL_{01} showed better performance but shared a similar pattern with the change of d_τ.

From our point of view, a value of d_τ from 40 to 60 m was optimal for the matching of 1:10k and 1:50k buildings. A smaller value of d_τ means higher *precision*. For example, when $d_\tau = 40$ m, RL_{02} obtained *Prec.* = 94.0%, *Recall* = 90.7% and $\kappa = 0.900$; RL_{03} obtained *Prec.* = 94.5%, *Recall* = 90.5% and $\kappa = 0.902$ (see their performance with $d_\tau = 50$ m in Table 2). Note that, it is less informative to rely merely on the *kappa* index. In certain applications one need to balance between *precision* and *recall*. For instance, when propagating updates from larger to smaller scales in an MRDB system, higher *recall* might be preferred over higher *precision* in establishing the corresponding relationships. This could reduce errors of omission in the data updating process.

Another parameter A_τ controls the minimum area in calculating Eq. (17). This parameter, however, is relatively constant for a certain range of scale. In our case, buildings at 1:50k should be bigger than 400 m^2 (ground unit) to be readable (according to the map

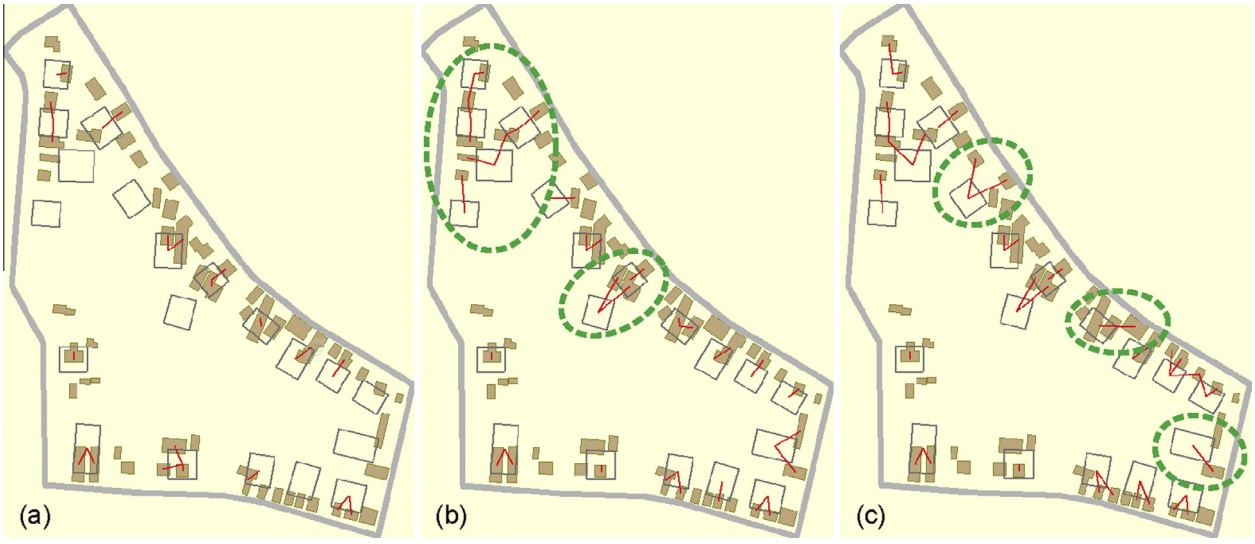


Fig. 9. Matching results in a highly ambiguous suburban area by (a) NB, (b) LW_{01}^{+} (highlighted are results obtained by considering relative positions) and (c) RL_{02} (highlighted are results obtained by considering relative orientations).

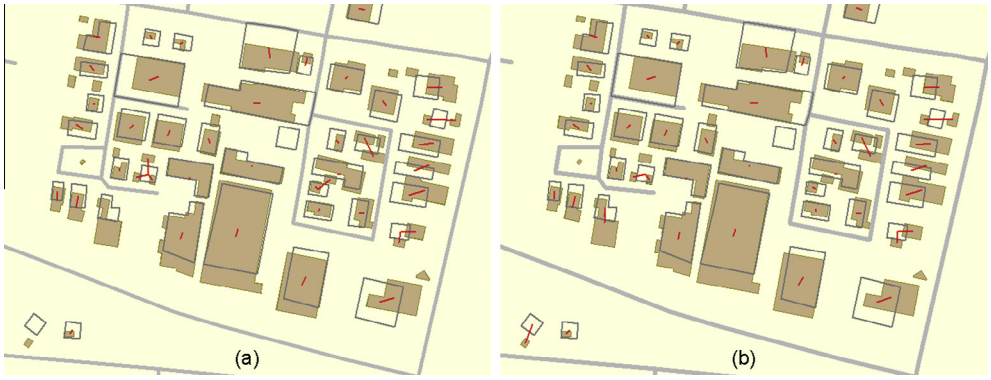


Fig. 10. Matching results in an industrial area by (a) NB and (b) RL_{02}.

Table 3
Performace of RL_{01} under different values of d_τ (m).

d_τ	tp	fp	fn	Prec. (%)	Recall (%)	κ
30	1756	122	252	93.5	87.5	0.875
40	1834	162	174	91.9	91.3	0.890
50	1863	212	145	89.8	92.8	0.885
60	1865	249	143	88.2	92.9	0.874
70	1853	281	155	86.8	92.3	0.861
80	1838	299	170	86.0	91.5	0.850
90	1817	309	191	85.5	90.5	0.840
100	1808	309	200	85.4	90.0	0.837

specifications of Kadaster), so small buildings at 1:10k have been enlarged to meet this specification. Therefore, A_τ is set to 400 m^2 and should not change for this scale range. In this way, domain-specific knowledge is incorporated into the matching algorithm. Our experiments show that such a treatment in Eq. (17) increased the matching performance. If a ratio between sizes of t_i and s_j without the treatment was compared, our approach gave poorer results (e.g. in this case RL_{02} obtained *Prec.* = 91.5%, *Recall* = 85.4% and $\kappa = 0.849$). The value of A_τ should be adapted if we are to match data for a different scale range. But the value can commonly be found in data specifications rather than being determined empirically.

3.2.4. Many-to-many correspondence and ambiguity reduction

Many-to-many correspondence is common in multi-scale data, which makes the matching more ambiguous. The explicit handling of many-to-many correspondence by Eq. (4) should improve the matching performance too. Fig. 11 shows how our approach deals better with many-to-many correspondence (including *n*-to-1 relationships). The situations highlighted in Fig. 11(a)–(c) are those better handled by RL_{02}; whereas those highlighted in Fig. 11(d)–(f) are *n*-to-*m* relationships not appropriately identified by NB.

To further analyze the improvement, we tested a one-to-one correspondence variant of our methods, which uses the original radial projection update rule (Eq. (3)). According to Parent and Zucker (1989), the original radial projection rule never saturates to unambiguous labeling (i.e. $p_{ij} = 1$). So in some situations the maximum p_{ij} for an object t_i can be less than 0.5. To get the best out of this variant, we postprocess the matching matrix after the relaxation by setting the max p_{ij} for each t_i to one. With this optimization, a one-to-one correspondence variant of RL_{02} obtained *Prec.* = 97.1%, *Recall* = 79.3% and $\kappa = 0.839$ with $d_\tau = 50$ m. This confirms that our methods (i.e. many-to-many correspondence variants) significantly improved the matching performance (cf. the performance of RL_{02} in Table 2).

To show how contextual information reduces ambiguities in the matching, typical situations are shown in Fig. 12. Clearly, NB cannot handle ambiguous situations due to a lack of contextual information (Fig. 12(e)–(h), (j) and (l)). The relaxation labeling process iteratively updates the matching matrix until a consistent result is reached (i.e. the relative relationships between matching pairs in a local neighborhood are compatible). This process finally reduces the ambiguities (Fig. 12(a)–(d), (i) and (k)).

The dynamic evolution of matching probabilities over iterations is shown in Figs. 13 and 14, which exemplifies how ambiguities were reduced. The two figures correspond to the situations in Figs. 12(i) and 12(k). In the first situation, the object #164 is initially matched to #217, and the second probable match is #222 (Fig. 13). The probabilities are initialized by NB. During the relaxation process, the probability of (#164, #197), which is initially very low, increases and becomes the final match due to the support it receives from neighboring pairs. The probability of the initial match (#164, #217) decreases because it is incompatible to other pairs. In the final state, the second probable match to #164 becomes #199. Although the probability does not exceed 0.5, (#164, #199) can be a reasonable match (see Fig. 12(i)). Similarly, #179 is matched to #222.

The second situation is highly ambiguous (Fig. 12(k)), #241 is initially matched to #177 but the matching probability drops quickly due to the lack of support (Fig. 14). After the relaxation, #224 and #234 are matched to #177, and #240 and #241 are matched to #183, which are acceptable. Note that the third probable matches to #177 and #183 are #241 and #235, respectively. These two possibilities seem to be also reasonable in Fig. 12(k). In addition, a close look at the probability trajectories shows that a higher order in the probability rank reflects a higher possibility of the match in reality.

A final note is that, although some ambiguous situations (e.g. Figs. 9 and 12(i) and 12(k)) are even difficult for humans, our approach can reduce the ambiguities and therefore, can find the most likely matches that are consistent in the local neighborhoods.

3.2.5. Initial probability assignment

As already mentioned previously, the matching matrix $P^{(0)}$ in Algorithm 1 is initialized by the NB-based probabilistic approach. It gave promising results, but it seems that our matching approach does not rely too much on the initial probabilities. This is evident in Figs. 13 and 14 that, the final probabilities are not correlated with their initial values.

To further investigate this problem, the matrix $P^{(0)}$ was initialized by randomization. This is done in three different ranges. In the first range $[0.1, 1.0]$, any t_i can be either or not matched to its candidates initially. In the second range $[0.1, 0.5]$, all t_i were initially not matched to its candidates. In the third range $[0.5, 1.0]$, all t_i were initially matched to its candidates. According to Eq. (1) or (2), a zero probability will not give a valid support. Therefore,

Fig. 11. Many-to-many correspondence is better handled in our approach: (a)–(c) are obtained by RL_{02} (solid circles indicate preferred n-to-m matches) ; (d)–(f) are the corresponding matches by NB (dashed circles indicate inappropriate matches).

we set the minimum initial probability to 0.1. The experiment was repeated for 10 times for each range. The results are exemplified in Table 4.

We can make the following observations from these experiments. First, for each initial range, the performance is stable (i.e. converging to the same result) no matter how many times the randomization was repeated. Second, the performance by different ranges of random initialization was similar. Third, although RL_{02} initialized by NB was better than the random initialization, the improvement was not significant. A detailed look at their matching results on the map did not show a substantial difference. So the conclusion here is that the accuracy of our matching approach does not rely critically on the initialization method. But one should note that a good estimation may give a (slightly) better result and, more importantly, requires fewer iterations to converge. For example, RL_{02} with a random initialization needed, on average, one and half as many iterations as that was required by the NB-based initialization.

This does not contradict the common view in computer vision that relaxation labeling techniques rely critically on the initialization method (Christmas et al., 1995; Zheng and Doermann, 2006; Sidibe et al., 2007). This is because our matching problem is different from a common computer vision problem. In computer vision, the target set is usually a (non-rigid) deformation of the model set, and the two are not placed in the same frame of reference (see e.g. Lee and Won, 2011). So in their approaches, the relaxation based matching and transformation are usually alternately iterated. The transformation is used to bring the two sets as close as possible iteratively. Spatial data, on the other hand, can always be roughly aligned with their spatial references. Therefore, the matching candidates always lie in some local neighborhood of the target object. This explains why the random initialization obtained reasonable results in our case.

3.2.6. Computational issues

The complexity of Algorithm 1 is very high. Several optimizing steps are taken in our approach. First, localization techniques are used to define the matching candidates of t_i (5) and the neighborhood structures (6) and (7). Hence the sizes of $C_i, \mathcal{N}_i$ and $\mathcal{N}_j$ are now statistically constant. For example, $\mathcal{N}_i$ is defined by DT which implies that $\mathrm{Exp}[|\mathcal{N}_i|] = 6$. Second, as most $p_{ij} \in P$ are equal to zero, we use a sparse matrix to store P, which largely reduces the storage need. Third, because the compatibility $r_{ij}(h,k)$ are evaluated repeatedly in the iteration, we firstly calculate the properties (e.g. size, orientation, shape, etc.), which are themselves time-consuming, and then store them as attributes. This significantly improved the efficiency. The efficiency can be further improved by storing all $r_{ij}(h,k)$ in a matrix (Sidibe et al., 2007), but this will introduce additional storage overhead.

In an experiment, we counted the number of iterations required for each street block and for each method, the box plot of iterations and the processing time are displayed in Fig. 15 (tests performed on a desktop with Core2 Duo CPU at 2.93G in the single task mode). It shows that the full processing time (indicated by •) is correlated to the number of iterations required to stop the relaxation. In general, methods with more relative relationships added required more iterations and longer processing time. More specifically, we found that matching in industrial areas needed less iterations, because the situations there are less ambiguous and the density of contextual information is higher. Besides, methods with the support function Eq. (2) (i.e. RL_{11}, RL_{12} and RL_{13}) converged a bit quicker than their counterparts with Eq. (1). This is because more contextual information can be involved in Eq. (2).

The box plot further suggests that most street blocks required relatively less iterations (less than 150), while there are a few outliers, which needed more than 300 iterations to stop. However, it is unlikely that every pair in such blocks requires this many steps to

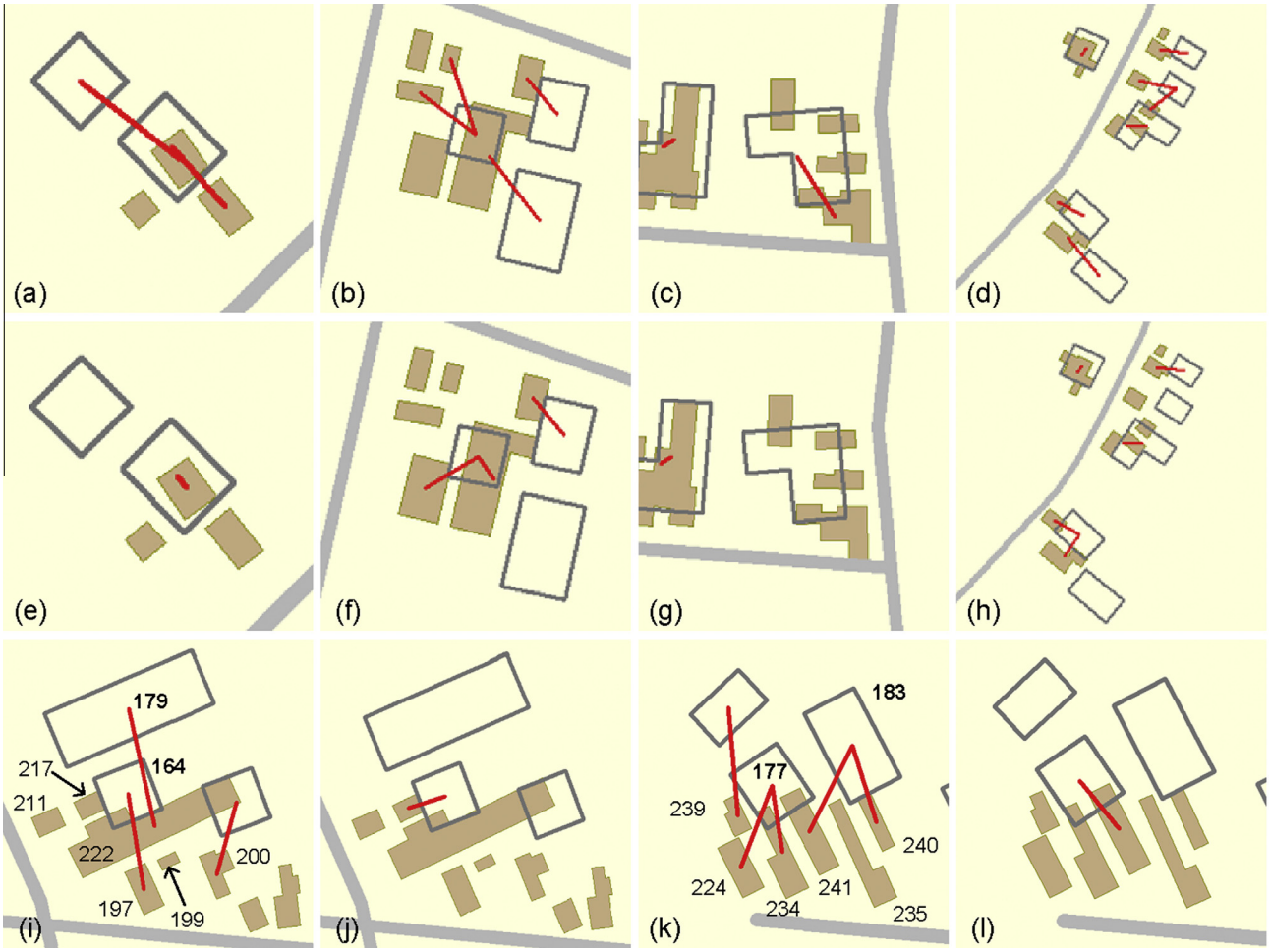


Fig. 12. Typical situations where ambiguities are reduced by our approach: (a)–(d), (i) and (k) are obtained by RL_{02}; (e)–(h), (j) and (l) are the corresponding matches by NB.

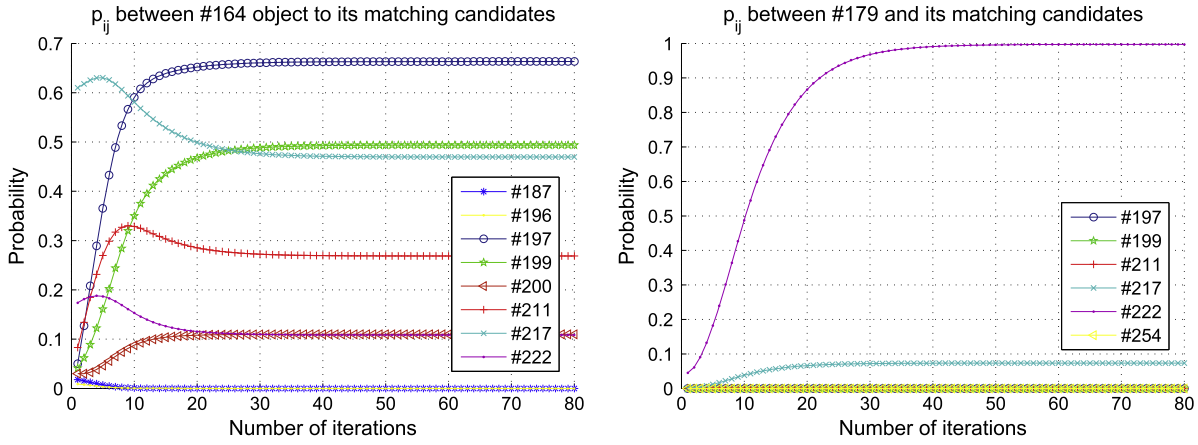


Fig. 13. Update of matching probabilities for the situation in Fig. 12(i).

converge. As is shown clearly in Figs. 13 and 14, many p_{ij} converged in some 20 steps and leveled off for the rest of the many steps. To give more insight, we counted the iterations needed by individual matching pairs and found that 96.7% of them converged within 20 steps. Many of them had to continue with the iteration because some pair in the same block needed much more steps to converge (i.e. $|\Delta p_{max}| < \Delta p_{stop}$), thus reducing the overall efficiency.

Hence in time-critical applications, one can sacrifice a little performance for speed. For example, when the process was stopped at the 150th step, RL_{02} obtained *Prec.* = 92.4%, *Recall* = 91.2% and $\kappa = 0.893$. This is almost the same to its performance in Table 2 but the processing time reduced from a full running time (approx. 10 *s*) to 7.5 s. The processing time of RL_{02} was further reduced (3.9 s) when it was stopped at the 20th step, with performance of *Prec.* = 92.6%, *Recall* = 90.9% and $\kappa = 0.892$. The gain in efficiency

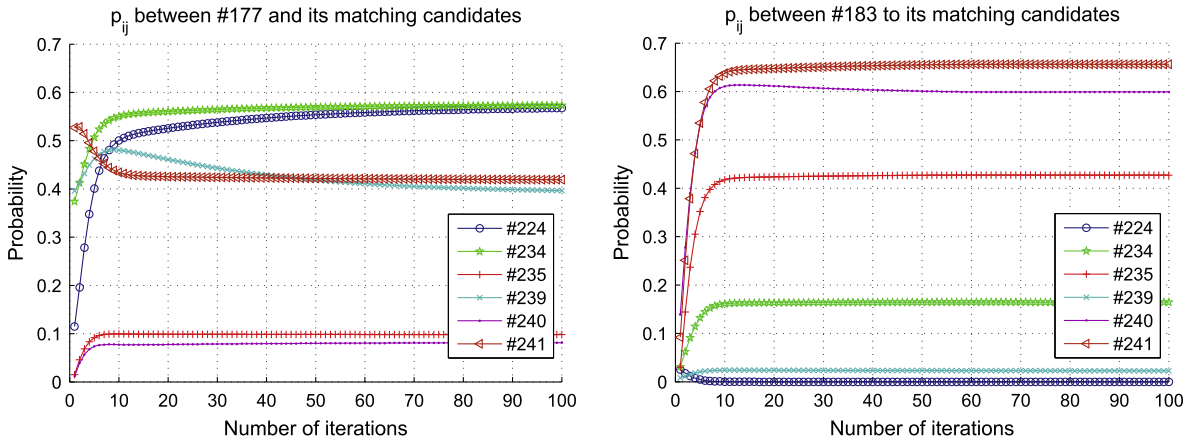


Fig. 14. Update of matching probabilities for the situation in Fig. 12(k).

Table 4
Performance of RL_{02} with random initialization ($d_\tau = 50$ m).

Initial range	*Prec.* (%)	*Recall* (%)	κ
[0.1, 1.0]	91.5	90.7	0.883
[0.1, 0.5]	92.3	90.1	0.885
[0.5, 1.0]	91.1	91.4	0.885

outweighs the loss in performance. The time needed by the various versions when stopped at the 150th and 20th step are plotted in Fig. 15 by ♢ and ◁, respectively. Another idea could be to limit the update to nodes with a significant change in the previous iteration. Yet, this is only useful after basic convergence, e.g., after ten or twenty iterations. Also, this will take up a bit more memory for keeping track of these nodes.

A final possibility to improve the efficiency is that, the proposed approach lends itself well to the parallel implementation. First, the relaxation labeling is carried out by parallel operations (Rosenfeld et al., 1976). That is, the update of p_{ij} at the $(t+1)$th iteration depends only on p_{ij} and p_{hk} of the tth iteration and the compatibility coefficient $r_{ij}(h,k)$ (see Eqs. (1)–(4)). Since the compatibility is constant for (t_i, s_j) with a given neighborhood definition, p_{ij} is updated independently from each other. Therefore, the sequence in which p_{ij} is updated is irrelevant. Second, since Algorithm 1 is applied independently to each street block, our approach could be further parallelized at the street block level.

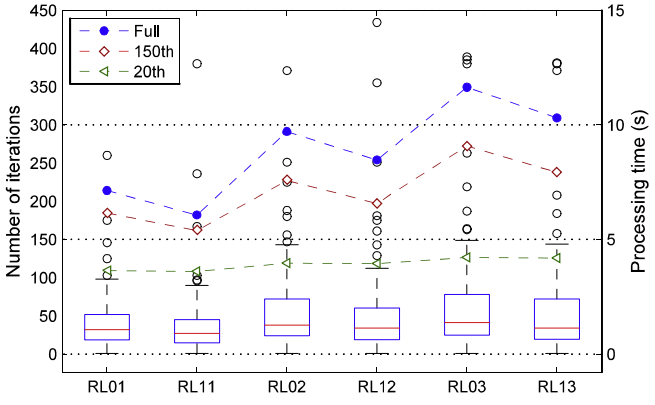


Fig. 15. Box plot of the number of iterations needed for the 105 street blocks and for different methods (○ indicates outliers in the number of iterations) and their processing time (● – full processing time, ♢ and ◁ – stopped respectively at the 150th and 20th iteration).

3.2.7. Remaining issues

Although our approach largely improves the matching and gives promising results, there is still room for improvement. This section identifies several common problems that could be further reduced (Fig. 16).

First, in some situations many s_j have equal chances to be matched with t_i (Fig. 16(a)), which reduces their average matching probability. For example, the probabilities that #830 matched to #348, #365, #328, #354 and #329 are 0.498, 0.496, 0.446, 0.426 and 0.351, respectively. Our current selection scheme, i.e., $p_{ij} > 0.5$, fails to identify these matches. By simply switching to a one-to-one scheme does not solve this problem. With Eq. (3) (a one-to-one scheme) the probabilities of these pairs are 0.225, 0.224, 0.201, 0.192 and 0.158. This is because the *radial update scheme* never saturates to $p_{ij} = 1$ (Parent and Zucker, 1989). Although other one-to-one mapping should find the most probable match such as (#830, #348), this is not the expected result (i.e. #830 should be matched to #348, #365, #328 and #354).

Another way could be to model one-to-one mapping independently, without respecting the constraint $\sum_{j=1}^{J} p_{ij} = 1$ (such as the NB-based matching). This would result in many-to-many correspondence, but in general it gave poorer matching results as we have shown. Our approach with Eq. (4) is getting closer. In the future it is possible to firstly identify highly ambiguous situations by analyzing the output probabilities and secondly to find the objects that are more likely matched to the target object using domain knowledge. This will further increase the *recall* of the matching.

Second, impossible links may be found by our approach. For example, two target objects are matched to two source objects in Fig. 16(b) but three links are created, which is not possible. This can be fixed by looking into the probabilities. The probability of $(\#448, \#700)$ is 0.620, which is lower than that of $(\#448, \#699)$ (0.784) and $(\#450, \#700)$ (0.871). Hence the link between #448 and #700 should be dropped. This will further increase the *precision* of the matching.

Third, the primary match is correctly found, but other matches can be missing in special cases. This usually occurs if several objects have been aggregated but one of them is more similar to the target object. For example, the probabilities that #670 matched to #648, #643, #641 and #639 are 0.861, 0.369, 0.289 and 0.198, respectively (Fig. 16(c)). This is a limitation of the relaxation labeling based matching, as contextual information does not help too much. Such situations can be easily matched using a different paradigm, e.g. by set-based analysis (Tversky, 1977; Winter, 2000). But the question is how to identified these situations from the result of a relaxation-based matching, such that the power of different approaches can be combined. An idea would be to analyze all

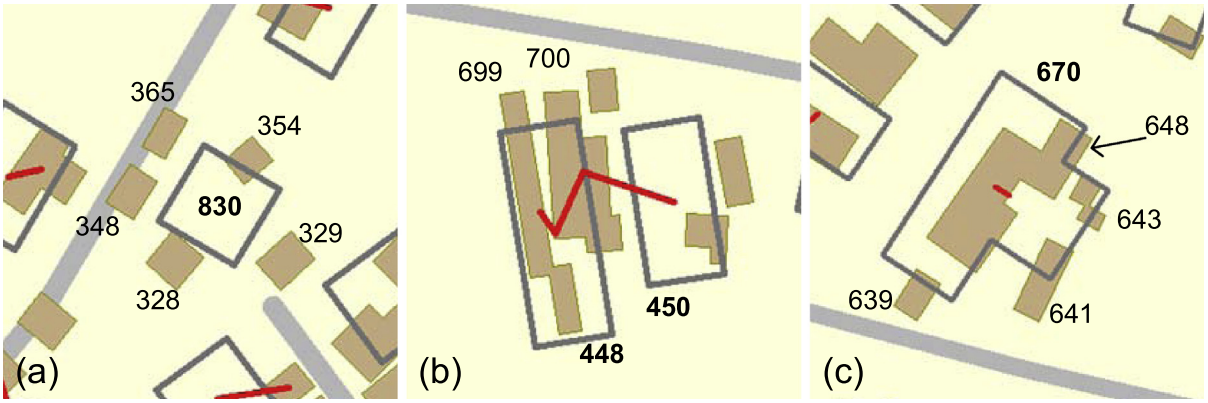


Fig. 16. Common issues identified for further improvement.

matches in a postprocessing stage. If a pair of objects differs dramatically in size or shape, one possibly identifies such a situation. But this deserves further investigation.

4. Conclusions and outlook

This paper describes a spatial data matching approach based on relaxation labeling techniques, which is an important step towards automated and more intelligent integration, evaluation and update of spatial data from various sources. The major contributions of our approach are twofold. First, we propose a domain-specific extension to standard relaxation techniques to explicitly model many-to-many correspondence in multi-scale spatial data. Such an extension works well in our domain, though it is not generally applicable because we do not handle the more general many-to-many correspondence that may be encountered in other domains. Second, we described a new compatibility coefficient suitable for objects that have not only positions but also lengths and widths. The use of relative relationships (i.e. relative positions, orientations, sizes and shapes) between spatial objects in quantifying the compatibility proves to be important in the matching task. In addition, the definition of local neighborhood structures leads to a more efficient implementation of the algorithm. In matching building features, our contextual approach showed superior performance in general and especially in ambiguous situations (e.g. suburban areas), in which many-to-many correspondence were better handled.

Some measures in the compatibility coefficient are specific to buildings, e.g., the orientation measure. They should be adapted if our approach is to be applied to a different geographic feature. The compatibility coefficient can be further extended by exploiting more relative relationships in the context. In the experiments, matching between 1:10k and 1:50k data was tested. Clearly, our approach can be applied for a different scale range. But further investigation should also be carried out to see which relative relationships still contribute to the matching for another scale range. This is because some relationships may no longer be preserved as the scale becomes smaller. Hence different relative relationships in the compatibility coefficient may have a different impact and thus should be finer-tuned for different scale ranges. Besides, our approach can be directly applied to matching spatial data of the same (or similar) scale. In this case the special treatments for matching multi-scale data such as many-to-many correspondence are not strictly followed.

In our experiments, we excluded built-up areas to better demonstrate the main problem. To apply our approach in a real-world application, built-up areas should also be linked to their larger scale footprints. An idea to automate this process is firstly to identify built-up areas (semantic or geometric criterion can be useful) and then to perform the matching using a polygon overlap-based analysis. This then can be combined with the results obtained in this paper.

Acknowledgements

We thank the national mapping agency, Kadaster of the Netherlands, for providing data to this work. This work was financially supported by National Natural Science Foundation of China (Grant No. 41301410), The National High Technology Research and Development Program of China (Grant Nos. 2012AA12A404 and 2012BAJ22B02-01), and China Postdoctoral Science Foundation funded project (Grant No. 2013M531742). The three anonymous reviewers whose comments substantially improved this paper are gratefully acknowledged.

Appendix A. Theoretical background concerning Eqs. (3) and (4)

This theoretical background follows the way given by Parent and Zucker (1989). Let p_{ij} be the probability that $s_j \in S$ may be matched to $t_i \in T$. The values of p_{ij} are restricted to the range [0, 1].

In vector notation, the probability vector for t_i is $\vec{p}_i$, which is a vector of probabilities that all s_j's, for $j \in [1,J]$, match to t_i. Hence the constraint $\sum_{j=1}^{J} p_{ij} = 1$ enforced in Eq. (3) can be written as $\vec{p}_i \cdot \mathbf{1} = 1$, where $\mathbf{1}$ is a J dimensional vector of 1's $(1,\ldots,1)$. Likewise, the support t_i receives can be denoted as a J dimensional vector $\vec{q}_i$, too. Both $\vec{p}_i$ and $\vec{q}_i$ are nonnegative, so they lie inside the positive quadrant of a vector space (e.g. Fig. A.17(a)).

Let $\vec{p}_i^{(t)}$ and $\vec{q}_i^{(t)}$ be the probability vector and the support vector of t_i at iteration t. The radial projection rule updates the probability vector by adding $\vec{p}_i$ and $\vec{q}_i$ and scaling them in a J dimensional vector space:

$$\vec{p}_i^{(t+1)} = \frac{\vec{p}_i^{(t)} + \vec{q}_i^{(t)}}{(\vec{p}_i^{(t)} + \vec{q}_i^{(t)}) \cdot \mathbf{1}} = \frac{\vec{p}_i^{(t)} + \vec{q}_i^{(t)}}{1 + \sum_{j=1}^{J} q_{ij}^{(t)}}. \tag{A.1}$$

This is a vector version of Eq. (3). The above equation is such that it satisfies the one-to-one constraint: $\sum_{j=1}^{J} p_{ij} = 1$ (i.e. $\vec{p}_i \cdot \mathbf{1} = 1$). In the iterative updating, the probability vector takes small steps until it coincides with the direction of the support vector. The denominator serves only as a scaling factor that does not change the direction of the updating in that space. The probability vector $\vec{p}_i$ always converges on a hyperplane ($\vec{p}_i \cdot \mathbf{1} = 1$) in the vector space due to the use of the above scaling factor. This is illustrated in Fig. A.17(a). A detailed account can be found in Parent and Zucker (1989).

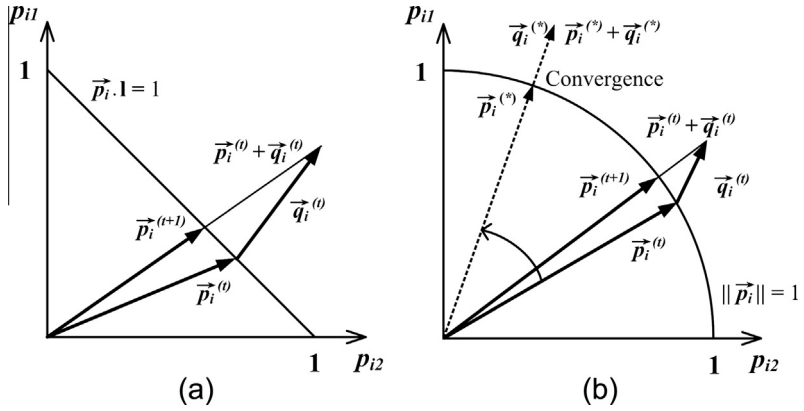


Fig. A.17. For the case of t_i matched with two potential candidates s_1 and s_2, updated matching probabilities p_{i1} and p_{i2} are obtained by scaling $\vec{p}_i^{(t)} + \vec{q}_i^{(t)}$ to satisfy constraints: (a) $\vec{p}_i \cdot \mathbf{1} = 1$ and (b) $\|\vec{p}_i\| = 1$.

Eq. (4) differs from Eq. (3) only in the scaling factor. To enforce many-to-many correspondence in our matching task, a necessary constraint to satisfy could be $\sum_{j=1}^{J}(p_{ij})^2 = 1$. This constraint allows that more than one p_{ij} may be larger than 0.5. In vector notation, this rewritten constraint can be expressed as $\|\vec{p}_i\| = 1$. This means that $\vec{p}_i$ should converge on a hyper-sphere instead of a hyperplane (as shown in Fig. A.17(b)). To do this, we choose a different scaling factor and rewrite Eq. A.1 as:

$$\vec{p}_i^{(t+1)} = \frac{\vec{p}_i^{(t)} + \vec{q}_i^{(t)}}{\|\vec{p}_i^{(t)}\| + \|\vec{q}_i^{(t)}\|} = \frac{\vec{p}_i^{(t)} + \vec{q}_i^{(t)}}{1 + \sqrt{\sum_{j=1}^{J}\left(q_{ij}^{(t)}\right)^2}}. \tag{A.2}$$

In the above equation, we scale the sum of $\vec{p}_i$ and $\vec{q}_i$ by the sum of their lengths, where $\|\vec{p}_i\| = 1$ and $(\sum_{j=1}^{J}(q_{ij}^{(t)})^2)^{1/2}$ is the length of $\vec{q}_i$ in a J dimensional vector space. This scaling factor provides a reasonable approximation. Although it projects $\vec{p}_i$ approximately onto the hyper-sphere in the first few iterations, as $\vec{p}_i$ approaches to the direction of the final $\vec{q}_i^{(*)}$ (see Fig. A.17(b)), the scaling factor $\|\vec{p}_i\| + \|\vec{q}_i\|$ will project $\vec{p}_i + \vec{q}_i$ exactly on the hyper-sphere ($\vec{p}_i^{(*)}$).

We empirically confirmed that $\|\vec{p}_i\| = 1$ holds for our matching results. Note that although such an extension works well for our application domain, it is not a universally applicable extension. To handle the more general case, one has to model arbitrary relationships between $\{t_i\}$ and $\{s_j\}$, which is out of the scope of this paper.

References

Balley, S., Parent, C., Spaccapietra, S., 2004. Modelling geographic data with multiple representations. Int. J. Geogr. Inf. Sci. 18 (4), 327–352.

Bard, S., 2004. Quality assessment of cartographic generalisation. Trans. GIS 8 (1), 63–81.

Beeri, C., Doytsher, Y., Kanza, Y., Safra, E., Sagiv, Y., 2005. Finding corresponding objects when integrating several geo-spatial datasets. In: Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems. ACM, New York, NY, USA, pp. 87–96, ISBN 1-59593-146-5.

Belongie, S., Malik, J., Puzicha, J., 2002. Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. 24 (4), 509–522.

Buttenfield, B., Delotto, J., 1989. Multiple Representations: Scientific Report for the Specialist Meeting, Tech. Rep. 89-3, State University of New York National Center for Geographic Information and Analysis, Buffalo, NY.

Christmas, W.J., Kittler, J., Petrou, M., 1995. Structural matching in computer vision using probabilistic relaxation. IEEE Trans. Pattern Anal. Mach. Intell. 17 (8), 749–764.

Chui, H., Rangarajan, A., 2003. A new point matching algorithm for non-rigid registration. Comput. Vis. Image Understanding 89 (23), 114–141.

Duchêne, C., Bard, S., Barillot, X., Ruas, A., Trévisan, J., Holzapfel, F., 2003. Quantitative and qualitative description of building orientation. In: 5th Workshop on Progress in Automated Map Generalization, Paris, 10, URL http://generalisation.icaci.org/images/files/workshop/workshop2003/duchene_et_al_v1.pdf.

Girres, J.-F., Touya, G., 2010. Quality assessment of the French OpenStreetMap dataset. Trans. GIS 14 (4), 435–459.

Hummel, R., Zucker, S., 1983. On the foundations of relaxation labelling processes. IEEE Trans. Pattern Anal. Mach. Intell. 5 (3), 267–287.

Kilpeläinen, T., 2000. Maintenance of multiple representation databases for topographic data. Cartogr. J. 37 (2), 101–107.

Kim, J., Yu, K., Heo, J., Lee, W., 2010. A new method for matching objects in two different geospatial datasets based on the geographic context. Comput. Geosci. 36 (9), 1115–1122.

Koukoletsos, T., Haklay, M., Ellul, C., 2012. Assessing data completeness of VGI through an automated matching procedure for linear data. Trans. GIS 16 (4), 477–498.

Lee, J.-H., Won, C.-H., 2011. Topology preserving relaxation labeling for nonrigid point matching. IEEE Trans. Pattern Anal. Mach. Intell. 33 (2), 427–432.

Lindeberg, T., 1994. Scale-Space Theory in Computer Vision. Kluwer Academic, Boston, USA.

Lowe, D.G. 1999. Object recognition from local scale-invariant features. In: International Conference on Computer Vision, Corfu, Greece, pp. 1150–1157

Mackaness, W.A., Ruas, A., 2007. Evaluation in the map generalisation process. In: Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (Eds.), Generalisation of Geographic Information: Cartographic Modelling and Applications, Series of International Cartographic Association. Elsevier Science, Amsterdam, pp. 89–111.

Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27 (10), 1615–1630.

Mustière, S., Devogele, T., 2008. Matching networks with different levels of detail. Geoinformatica 12 (4), 435–453.

Parent, P., Zucker, S., 1989. Radial projection: an efficient update rule for relaxation labeling. IEEE Trans. Pattern Anal. Mach. Intell. 11 (8), 886–889.

Pelillo, M., 1997. The dynamics of nonlinear relaxation labeling processes. J. Math. Imag. Vis. 7 (4), 309–323.

Raimond, A.-M.O., Mustière, S., 2008. Data matching – a matter of belief. In: Headway in Spatial Data Handling. Lecture Notes in Geoinformation and Cartography. Springer, pp. 501–519.

Ranade, S., Rosenfeld, A., 1980. Point pattern matching by relaxation. Pattern Recogn. 12 (4), 269–275.

Rosenfeld, A., Hummel, R.A., Zucker, S.W., 1976. Scene labeling by relaxation operations. IEEE Trans. Syst., Man, Cybernet. 6 (6), 420–433.

Ruiz, J.J., Ariza, F.J., Ureña, M.A., Blázquez, E.B., 2011. Digital map conflation: a review of the process and a proposal for classification. Int. J. Geogr. Inf. Sci. 25 (9), 1439–1466.

Saalfeld, A., 1985. A fast rubber-sheeting transformation using simplical coordinates. Am. Cartographer 12 (2), 169–173.

Saalfeld, A., 1988. Automated map compilation. Int. J. Geogr. Inf. Sci. 2 (3), 217–228.

Samal, A., Seth, S., Cueto, K., 2004. A feature-based approach to conflation of geospatial sources. Int. J. Geogr. Inf. Sci. 18 (5), 459–489.

Sidibe, D., Montesinos, P., Janaqi, S., 2007. Fast and robust image matching using contextual information and relaxation. In: Ranchordas, A., Araújo, H., Vitrià, J. (Eds.), Proceedings of the 2nd International Conference on Computer Vision Theory and Applications, Barcelona, pp. 68–75.

Song, W., Keller, J.M., Haithcoat, T.L., Davis, C.H., 2011. Relaxation-based point feature matching for vector map conflation. Trans. GIS 15 (1), 43–60.

Steiniger, S., Weibel, R., 2007. Relations among map objects in cartographic generalization. Cartography Geogr. Inf. Sci. 34 (3), 175–197.

Stoter, J., Burghardt, D., Duchêne, C., Baella, B., Bakker, N., Blok, C., Pla, M., Regnauld, N., Touya, G., Schmid, S., 2009a. Methodology for evaluating automated map generalization in commercial software. Comput., Environ. Urban Syst. 33 (5), 311–324.

Stoter, J., van Smaalen, J., Bakker, N., Hardy, P., 2009b. Specifying map requirements for automated generalization of topographic data. Cartogr. J. 46 (3), 214–227.

Tong, X., Shi, W., Deng, S., 2009. A probability-based multi-measure feature matching method in map conflation. Int. J. Remote Sens. 30 (20), 5453–5472.
Tversky, A., 1977. Features of similarity. Psychol. Rev. 84 (4), 327–352.
Volz, S., 2006. An iterative approach for matching multiple representations of street data. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVI-2-W40, Hannover, pp. 101–110.
Walter, V., Fritsch, D., 1999. Matching spatial datasets: a statistical approach. Int. J. Geogr. Inf. Sci. 13 (5), 445–473.
Wang, H., Hancock, E.R., 2008. Probabilistic relaxation labelling using the Fokker-Planck equation. Pattern Recog. 41 (11), 3393–3411.
Winter, S., 2000. Location similarity of regions. ISPRS J. Photogramm. Remote Sens. 55 (3), 189–200.
Yang, B., Zhang, Y., Luan, X., 2013. A probabilistic relaxation approach for matching road networks. Int. J. Geogr. Inf. Sci. 27 (2), 319–338.
Zhang, M., Meng, L., 2007. An iterative road-matching approach for the integration of postal data. Comput., Environ. Urban Syst. 31 (5), 597–615.
Zhang, X., Zhao, X., Molenaar, M., Stoter, J., Kraak, M.-J., Ai, T., 2012. Pattern classification approaches to matching building polygons at multiple scales. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. I-2, Melbourne, pp. 19–24.
Zhang, X., Stoter, J., Ai, T., Kraak, M.-J., Molenaar, M., 2013. Automated evaluation of building alignments in generalized maps. Int. J. Geogr. Inf. Sci. 27 (8), 1550–1571.
Zheng, Y., Doermann, D., 2006. Robust point matching for nonrigid shapes by preserving local neighborhood structures. IEEE Trans. Pattern Anal. Mach. Intell. 28 (4), 643–649.