



# Level of Details Harmonization Operations in OpenStreetMap Based Large Scale Maps

Guillaume Touya, Matthieu Baley

## ► To cite this version:

Guillaume Touya, Matthieu Baley. Level of Details Harmonization Operations in OpenStreetMap Based Large Scale Maps. Leitner, Michael; Jokar Arsanjani, Jamal. Citizen Empowered Mapping, 18, pp.3-25, 2017, Geotechnologies and the Environment, 10.1007/978-3-319-51629-5\_1 . hal-02170261

**HAL Id: hal-02170261**

**<https://hal.science/hal-02170261>**

Submitted on 1 Jul 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Level of Details Harmonization Operations in OpenStreetMap Based Large Scale Maps

Guillaume Touya, Matthieu Baley

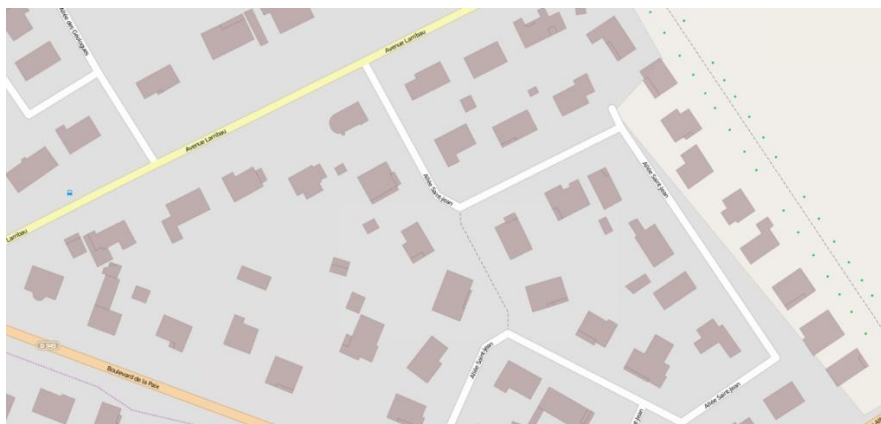
*IGN – COGIT, 73 avenue de Paris 94165, Saint-Mandé, France*

**Abstract** OpenStreetMap data comprise of very detailed (e.g. zebra crossing) and quite rough features (e.g. built-up area). But making large scale maps from data with inconsistent level of detail often blurs map comprehension. This paper explores the automatic harmonization of OpenStreetMap data for large scale maps, i.e. the process that transforms rough objects to make them consistent with detailed objects. A typology of the new operators that harmonization requires is presented and six algorithms that implement the operators are described. Experiments with these algorithms raise several research questions about automation, parametrization, or the level of abstraction of the transformation, which are discussed in the paper.

## Introduction

As OpenStreetMap (OSM) is growing larger every day, practical applications based on OSM data are flourishing, but the initial goal of the project was to produce open topographical maps. A quick look at the default map output provided by OSM shows that it is difficult to create good legible maps out of the huge amount of data in OSM. One of the main obstacles to the creation of good legible maps from OSM data is the heterogeneity of its level of detail (LoD). For example, in the database, very detailed objects (e.g. zebra crossings) coexist with raw objects (e.g. shorelines extracted from Landsat imagery). This heterogeneity is troublesome for small scale maps, as detailed objects should be removed or simplified, but also for large scale maps, as rough objects are often inconsistent with detailed features of the map (Touya 2012a). Regarding small scale maps, it can be considered as a map generalization problem. Although there is little research effort on generalizing OSM data (Klammer 2013, Schmid and Janetzek 2013, Sester et al. 2014), this paper focuses instead on legibility problems caused by the level of detail heterogeneity in large scale maps, i.e. scales larger than 1:25,000. In such large scale maps, detailed objects can be displayed in the map without any generalization transformation, because the scale ratio makes map symbols close to their real extent on the ground. But rough objects are also included in the map, because it is often better to display rough information than

display nothing all. For instance, if all forest polygons are rough, it is better to include them in the map rather than leaving them out. The first step to make such large scale maps is to find a way to infer the level of detail of OSM data in order to discriminate detailed objects from rough objects (Touya and Brando 2013, Touya and Reimer 2015). The understanding of a map is highly dependent on the way the reader grasps spatial relations between map objects. As a consequence, the level of detail inconsistencies are mainly damaging when occurring between spatially related objects. For instance, the gray built-up area in Figure 1 does not include some of the town buildings on the right. The map reader may thus be troubled and may misinterpret what the built-up area is.



**Fig. 1.** Buildings on the right should be inside the built-up area (in gray) to make it a understandable spatial relation

Dealing with LoD inconsistencies in large scale maps can be seen as a new automatic mapping process, namely to transform rough objects to make them consistent with detailed objects when both types of objects share a spatial relation that helps to understand the map. We call such a process harmonizing level of detail, implying that the harmonization increases LoD. It should be noted that generalizing heterogeneous data is also a kind of harmonization but it is not the focus of this paper. The automation of harmonization raises two questions. Is it possible to automatically harmonize OSM maps? Is it meaningful to transform data without any additional information from ground truth to make it more detailed? The work presented in this paper seeks to explore both questions by experimenting first attempts of automatic harmonization on OSM data.

After the introduction, the following second section of this paper, briefly discusses the notions of level of detail, scale and quality, and briefly describes a method to infer the level of detail, as a first step for harmonization. The third part precisely defines cartographic LoD harmonization and proposes a typology of possible harmonization operators. Section four is the core of the paper and

describes six algorithms to harmonize different types of LoD inconsistencies. The fifth part discusses several issues on the automation and the meaning of harmonization for a map. The last section draws conclusions and explores further research.

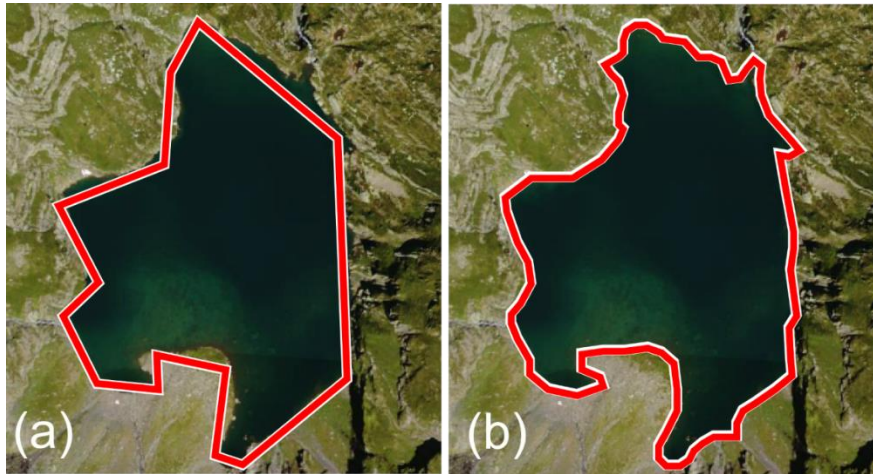
## Level of Detail and Data Quality

The scale of a map is the mathematical ratio between a distance measured in the map and the same distance measured on the ground. But the scale is not only a ratio, it is also closely related to the content of the map and its resolution (Mackaness 2007). Indeed, the scale limits the map to a certain extent, and the human perception limitations bound what can be displayed on the map. On the other hand, geographical information databases with vector data can be zoomed in and out, thus they cannot be defined by a single scale. So, we usually refer to the level of detail (LoD) when we want to define the resolution, or the granularity of a geographical database. Unlike scale, the level of detail is not a mathematically defined notion, and the fuzziness of its definition may make it hard to assess. In previous work, we defined the level of detail as a complex notion (Touya and Brando 2013) that encapsulates elements of:

- conceptual schema (a tree representation is more detailed than a forest),
- attribute resolution,
- geometric resolution, i.e. smallest length between two vertices (Figure 2),
- geometric precision or accuracy,
- granularity (size of the smallest detail of geometries).

OSM can be considered as a large geographical database, where the LoD of features varies as the contributors with varying skills may use sources of varying scales or resolutions.

There is a substantial literature on all aspects of OSM data quality (Haklay 2010, Girres and Touya 2010, Mooney and Corcoran 2012). Although the level of detail comprises of some elements of data quality, the aim here is not exactly to assess data quality. A raw lake outline (i.e. with few vertices, see Figure 2a) can be considered as a bad quality feature if it is expected to have a high level of detail, or conversely a good quality feature if the expected level of detail is not so high (if the aim is to make a map at a small scale, for instance). In this work, the focus is not on quality, precision or accuracy alone, so an inaccurate position will only be considered as a (major) factor for a low level of detail.



**Fig. 2.** Two captures of the same lake at different levels of detail

Scale can be inferred from the geometrical resolution and the analysis of similar features in existing maps (Reimer et al. 2014). For instance, Biljecki et al. (2014) proposed several metrics to infer LoD in 3D city models. Regarding OSM, the LoD inference is automatically possible using multiple criteria decision techniques (Touya and Brando 2013, Touya and Reimer 2015), where the used criteria correspond to different aspects of LoD (resolution, precision, etc.). Then, the LoD inconsistencies can be identified by searching for key anomalous, or improbable spatial relations between detailed and rough features. For instance, trees should not be located on roads, or land use parcels should not extend over coastlines (Touya and Brando 2013). Improvements and alternative methods are, of course, necessary to get a better inference of the individual level of detail of OSM data (Touya and Reimer 2015), but this is not the focus of the presented work. As a consequence, the results of the inference method from Touya and Reimer (2015), i.e. the classification of OSM features into one of five LoD categories from *street* LoD to *country* LoD, are used as inputs for the harmonization methods presented in the next section.

The problem of LoD inconsistencies is not specific to the derivation of maps from OSM data, but may occur with any other volunteered geographical information. However, we choose to focus on OSM as the project is complete enough to derive large scale maps with a high density of information in areas with many contributors, such as, from Western Europe. OSM also contains data for every part of the world, making processes to derive maps automatically useful for many people.

## Cartographic LoD Harmonization

### *Problem Statement*

We define cartographic LoD harmonization as the mapmaking process that transforms features involved in a LoD inconsistency, in order to make the map more legible and comprehensible. When the target is a small scale map, or a map where the LoD of the rough feature of the inconsistency matches the scale, harmonization can be brought down to map generalization. But, in this paper, we only focus on large scale target maps, where simplification is not necessary and harmonization is a new problem. Moreover, the aim is not to provide quality control for the OSM dataset. When problematic LoD inconsistencies are identified, it is the readability that guides the transformation and not the quality control. In this case, the transformation should be a balance between position preservation and map legibility. As a result, caricature operations may be preferred to transformations guided by ground truth. OSM data are supposed to be more or less incomplete with some objects that could be in the data, but have not yet been captured. So, harmonization operations should take into account that the possible lack of detail may be due to incompleteness rather than LoD. For instance, when an object should be near a road and it is not, then the inconsistency may be caused by the inaccuracy of the object location, or by the absence of a road that exists in the real world. The key to harmonize LoD inconsistent spatial relations in the map is to transform the rough counterpart of the spatial relation while preserving the detailed member.

Research in cartographic generalization and multiple representation databases already focused on relations between geographical features at different LoDs. In multiple representation databases, spatial relations can be horizontal or vertical (Mustière and Moulin 2002, Burghardt et al 2010). Horizontal relations involve features with a similar level of detail, such as a building located at the end of a dead end road, while vertical relations involve two features at different LoDs that represent the same real world feature, such as a city represented by a polygon or a point. Spatial relations involved in LoD inconsistencies are neither vertical, nor horizontal. Following the same vocabulary, we can consider them as diagonal relations.

There is no way to transform rough features into detailed and accurate abstractions of real world entities they represent, so, in a way, harmonization aims to make the map more readable. As stated by Monmonier (1996), “not only it is easy to lie with maps, it is ESSENTIAL.” Maps are systems of relationships (Mackaness, Burghardt and Duchêne 2014), which means that most of the meaning of the map is conveyed by relations, and preserving relations by making them more legible improves the way maps are understood. However, transforming map features too much without safeguard can damage map readability more than

it improves it. Then, our safeguard is the evaluation of harmonization to verify that map features are not too much transformed.

### ***Typology of Harmonization Operators***

In order to derive legible maps from LoD inconsistent information, harmonization requires operation that can be related to other automated cartography processes, such as cartographic generalization or text placement. Typologies of generalization operators (e.g. simplification, displacement, elimination) already exist, see for instance the ones from Foerster, Stoter and Kobben (2007) and Regnauld and McMaster (2007). In our case, we preferred referring to a broader typology of operators for multiscale maps from Roth, Brewer and Stryker (2011). The following operators are introduced for harmonization as derivatives of Roth et al.'s operators:

- Merge/Dissolve
- Adjust Shape
- Displace and enhance
- Disambiguation

**Merge** is defined by Roth et al. (2011) as a “*replacement of a feature with a representative feature of equal dimensionality*” and illustrated by a group of small islands merged to the nearby big island. Regarding harmonization, the merge operation is useful to improve the geometry of a feature by merging detailed features that should be part of the rough feature. Geographical datasets often comprise high level objects that are aggregates of lower level objects of the dataset. For example, a city is an aggregate of buildings, roads, and parks. In OSM, such aggregate objects are very common and are generally less detailed than their components. This generates the most frequent LoD inconsistencies with obvious components that lie just outside the aggregate (see section 4.1). But merge is also coupled with **dissolve**, because the detailed features might sometimes be dissolved from the rough feature rather than merged, when they should not be part of the aggregate. For instance, a primary road cannot be part of a school site (see section 4.2).

**Adjust shape** is the adjustment of a least detailed shape without changing feature dimensionality, to avoid an improbable relation. It derives from the Roth et al. (2011) Adjust shape operator that was only dedicated to symbols and not to feature geometry. Modifying the shape of a lake in order to avoid intersections with roads is an example of the adjust shape operator (see section 4.4).

**Displace and enhance** is a displacement of a map feature in order to preserve or emphasize a relation. It is a mix of the “displace” and “enhance”

operators from Roth et al. (2011). The displacement of trees along roads to enable a real alignment of tree symbols is an example of the “displace and enhance” operator (see sections 4.3 and 4.5).

Sometimes, the map reader does not know if the improbable relation is true or if there is a problem of data quality (e.g. missing features). **Disambiguation** is a new operator that aims at the removal of this ambiguity in map reading without consideration for the ground truth (that we do not know). It arbitrarily removes the improbable spatial relation. For instance, when a group of close buildings is inside a forest without clearing, the addition of the clearing is a disambiguation operation (see section 4.6). Finally, harmonization also has to make use of existing operators such as displacement to correct inaccurate positions.

## Examples of Algorithms for Harmonization Operations

This section describes several algorithms (**Erreur ! Source du renvoi introuvable.**) that implement the operations introduced in the previous section.

**Table 1.** Table summarizing algorithms described in this paper, with bold features in the inconsistency column represent the rough member of the relation.

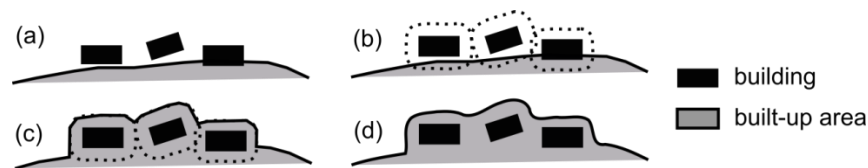
Algorithm	Operation	LoD inconsistency
Built-up area extension	Merge	Buildings that lie just outside a <b>built-up area</b>
Functional site adjustment	Merge	A <b>functional site</b> that excludes some of its component/ includes false components
Tree alignment along roads	Displace and enhance	<b>Trees</b> along a road but not aligned
Intersection removal	Adjust shape	Roads/paths intersecting a <b>lake</b>
Logical displacement	Displace and enhance	<b>Bus stops</b> too far from the nearest road
Clearing addition	Disambiguation	Building groups inside a <b>forest area</b>



## Built-up Area Extension

### Algorithm description

This algorithm seeks to extend the limit of built-up areas to include all buildings of the actual built-up area, i.e. buildings that lie just outside the limit, but also all buildings that are very close to these buildings. In this case, the only information available to draw a more detailed extent for the built-up area is that the components that caused the inconsistency should be inside the aggregate, and no longer outside. Several algorithms exist to compute the boundaries of built-up areas only using buildings (Boffet 2000, Chaudhry and Mackaness 2008, Walter 2008). We propose to use a similar strategy based on buffering the buildings, but in an iterative way. At each step, the algorithm searches buildings that are within a radius around the built-up area, but not yet inside the area. The radius used is 15 m, derived from Boffet (2000). Figure 3 shows the three steps of the algorithm, at each iteration: (i) buffers, with the same radius, are computed around the components to include, (ii) buffers are merged to the built-up area, and (iii) the outline is simplified to get a consistent resolution all along, as far as possible. In order to avoid too large extensions, the radius is cushioned at each iteration, with a 0.8 factor. The algorithm stops when there is no new building found within the search radius.



**Fig. 3.** Three steps of one iteration of the algorithm to extend built-up areas. (a) Initial state, (b) compute buffers around buildings, (c) merge the built-up area with buffers, (d) simplify to preserve the resolution.

The boundary simplification algorithm is a simple Douglas and Peucker filter (1973), because the sharp shapes created by this simplification algorithm look like the shapes of initial rough built-up areas.

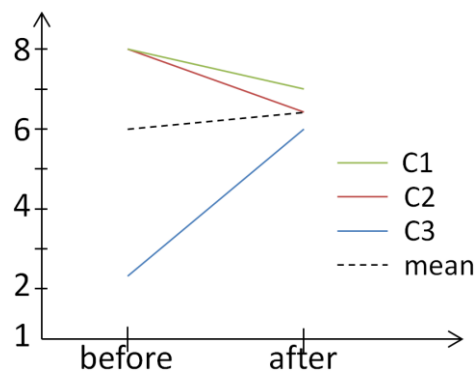
### Evaluation

Constraint-based evaluation is commonly used to evaluate maps derived by automatic generalization (Stoter et al. 2009, Touya 2012b). Constraints, such as *building area should not be below 0.4 mm<sup>2</sup> on the map*, are defined according to map specifications and known eye perception limits (Salichtchev 1983). Such constraints have been defined to evaluate this harmonization algorithm, some for preserving the initial shape/granularity of features, and some for assuring that

inconsistencies are really removed. Three constraints are used to evaluate harmonization:

- (C1) A shape preservation constraint that uses a surface distance (Girres and Touya 2010) to measure that the general shape has not been distorted too much.
- (C2) A granularity preservation constraint to measure that vertex density, used as a proxy for granularity by Reimer et al. (2014), has not increased too much after harmonization.
- (C3) A constraint that counts the number of dilated buildings intersecting the built-up area outline. The constraint is fully satisfied when there is no intersection.

An instance, or a monitor (Touya 2012b), is created for each constraint and each built-up area object. The monitors assess the constraint satisfaction given the current geometry of the built-up area. In this evaluation, the satisfaction of the constraint is retrieved before and after the harmonization to verify that C3 satisfaction has increased, while the C1 and the C2 satisfaction remained stable. The constraint satisfaction is expressed from 1 (not satisfied at all) to 8 (perfectly satisfied), similarly to the generalization constraints from Touya (2012b). A test area was chosen in the south west of France comprising around 50 built-up areas of small/medium towns with LoD inconsistencies. The built-up extension algorithm is triggered automatically with parameters given in the algorithm description. Results show a significant increase of C3 satisfaction, while C1 and C2 satisfactions slightly decrease but remain satisfied (Figure 4). The mean of satisfactions only slightly increases, but the lack of an unsatisfied constraint after harmonization is a better indicator, and it proves the increased global quality of the harmonized map.



**Fig. 4.** Mean constraint satisfaction evolution during built-up extension harmonizations.

## Functional Site Adjustment

Regarding functional sites (Chaudhry et al. 2009, Mackaness and Chaudhry 2011), such as schools or hospitals, boundaries are often crisp and identified on the ground by buildings, walls, or barriers. So, in this case, the merge/dissolve operation should set more realistic bounds for the site, and as a consequence, an algorithm based on dilatation cannot be used. The principles of the proposed “functional site adjustment” algorithm is to infer the probability of features that are near the initial boundary, to be part of the functional site or not. Then, boundaries of the site are displaced around the included or excluded component, without introducing any gap.

First of all, a belonging function is defined for each type of site. The function uses semantics and topology to infer if an object is a component of the site, or not (Chaudhry et al. 2009). We decomposed the belonging function into two functions that are specialized for each type of functional site: A *functional belonging* that uses semantics, e.g. a library or a football field is more prone to be part of a school while a primary road is not; and a *spatial belonging* that uses topological and metrical measures, e.g. a building that is 99% inside the site is more likely to be a part of the site. Both belonging functions give negative values for unlikely partonomy and positive values for probable partonomy. These functions were implemented for two types of sites, namely schools and hospitals. For instance, schools functional belonging value is 3 for objects tagged as churches or sports fields, 10 for objects tagged as libraries, and -10 for buildings tagged as commercial, and -15 for roads tagged as highways (Table 2). The spatial belonging computes the percentage of overlap between both geometries with a positive value over 60% and a negative value below 30%. Then, functional and spatial belongings are summed and compared to two thresholds: A belonging threshold (2 was used for schools and hospitals) and an exclusion threshold (-2 was used). Between the thresholds, the boundary is not modified by the object tested.

**Table 2.** Bonuses and maluses affected by the function belonging of school components for specific tags of OSM.

Tag key	Tag value	Functional belonging
building	<i>any value</i>	1
building	dormitory	5
building	church	3
building	chapel	3
building	civic	3

building	commercial	-10
building	industrial	-10
building	residential	-15
amenity	school	10
amenity	university	10
amenity	college	10
amenity	library	10
leisure	pitch	5
sports	<i>any value</i>	5
highway	primary	-15
highway	secondary	-15
highway	tertiary	-10
highway	residential	-10

Finally, the boundary is adjusted to include or exclude the intersecting objects. When the object to include (or to exclude) has a polygonal geometry, a polygon union (or difference) is used with a smoothing of intersections, i.e. the vertices just before the intersections are removed from the geometry (see the zoomed image in Figure 5). When the object has a linear geometry, the vertices of the boundary between the intersections with the line are replaced with the vertices of the line between the intersections.

Figure 5 shows the example of a high school whose initial boundaries intersect with several buildings, with some buildings (at the bottom and on the left) that are included in the high school and one (on the right) that is excluded.



**Fig. 5.** High school with rough limits harmonized to be consistent with the detailed buildings and sports grounds

## *Tree Alignment Along Roads*

When the “displace and enhance” operator is required, we have to choose which feature is displaced to enhance the relation. In the case of trees along roads, trees are often poorly detailed because they are hard to capture precisely, which makes tree alignments often overlap with road symbols (Figure 6). Here, we propose to displace trees because their location is already inaccurate, and because their displacement does not cause many repercussions in terms of symbol overlap in the map. On the contrary, moving or distorting the road would require some propagation of the transformation to connected roads.

The algorithm displaces trees in order to reduce the overlap between symbols and enhances the relation by forcing the alignment to the road symbol (Figure 6). Tree alignments on the right and on the left of a road are first identified by computing left and right buffers on roads and counting the trees inside each buffer. Then, right and left offset lines (like buffers without the caps) are computed with a distance that is the addition of the road symbol width and half the tree symbol radius (to maintain some overlap that mimics the fact that tree branches might overlap with road pavement). Finally, trees are projected on the offset and moved apart a little, in case projections are too close to each other (the tree symbol diameter is used as a minimal distance). When a tree is along two roads (i.e. at a crossroad), which is detected by the fact that a tree belongs to the left and right side of two different roads, the projected position is the intersection of both offset lines.

Figure 6 shows some automatic results obtained at the 1: 5,000 scale of the French city of Bordeaux, which is quite a large city where trees have been captured extensively by OSM contributors. Hundreds of overlapping alignments have been automatically identified and successfully displaced. Some remaining problems have appeared when there are several rows of trees, because the row along the road sometimes overlaps with other rows inside a park or a square. This problem would require some displacement propagation, and this issue is discussed in section 5.2.



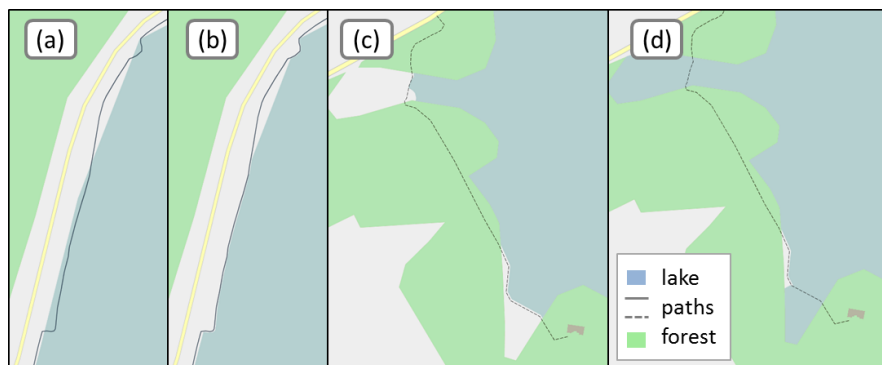
**Fig. 6.** a) Inaccurate trees overlapping a road. b) computation of an offset on the side of the road with trees. c) trees aligned on the offset line

### ***Intersection Removal***

Intersection removal is the modification of a linear or areal object that is rough, with a more detailed boundary, using other detailed objects that are in relation with the rough object, in order to figure out how the more detailed outline needs to be drawn. For instance, when objects like paths or buildings intersect the outline of a lake, a new outline is drawn avoiding such inconsistencies. Depending on the type of objects involved in the inconsistency, the intersection removal has to be made differently, and different implementations are possible, e.g. if space is required between the intersecting objects or if objects might be adjacent. In this paper, we describe an algorithm for paths or roads that cross lakes, so a space is required between the path and the lake boundary.

Lakes have bona fide boundaries (Smith and Varzi 2000), i.e. there is a physical discontinuity that marks the boundary on the ground. So, more information on the intersection removal can be deduced from the geographical characteristics of objects involved in the inconsistency. For instance, a detailed bicycle path (captured by GPS tracks) can intersect the outline of a rough lake captured on satellite images (Figure 7). The bicycle path has a certain width, so the harmonized lake outline cannot be adjacent to the bicycle path. Then, a small gap is added between the path and the lake, using an offset on the lake side of the path. The algorithm has the following steps: (i) Determine which side of the line

(given the order of the vertices) the main part of the lake is (i.e., the portion with the largest area); (ii) remove the portion of the lake that is on the other side of the line; (iii) define an offset of the line (i.e. a buffer without cap) only on the side where the main part of the lake is with a width related to the real width of the line (use some width information if available in the tags), plus a small gap (0.1 mm in the map), and remove the offset from the lake polygon; (iv) compute a cushioned version of the offset on the segments before and after the intersecting segments of the line to avoid sharp differences where the intersection has been removed (Figure 7).



**Fig. 7.** (a) A bicycle path intersecting a lake; (b) harmonized lake outline; (c) an example of bridges with bad harmonization; (d) harmonization with automatic detection of bridges

However, Figure 7 shows that, sometimes, paths crossing a lake are just bridges and there is no inconsistency that needs to be resolved. To avoid bad harmonization in such cases (Figure 7c), the intersection removal algorithm is improved with a pre-step that automatically identifies parts of a path that probably belong to a bridge. First, the algorithm checks the following semantics: If there is a tag “bridge” or “man\_made”=’bridge’ on the line, it is considered to be a bridge. Then, the two following characteristics of bridge sections are used for the automatic identification:

1. The middle of a bridge section is more “inside” the lake than either of its endpoints (is this what you want to say?) (Figure 8a and b),
2. The angle between the bridge section and the nearest lake shore is close to 90° (Figure 8c and d).

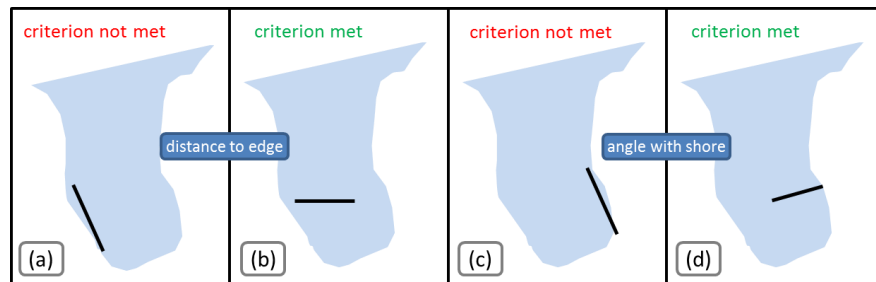


Figure 8. a) a segment whose middle point is as close to lake boundaries as ending points (criterion not met), b) a segment that meets distance to edge criterion, c) a segment not orthogonal to lake boundary (criterion not met), c) a segment orthogonal to lake boundary (angle shore criterion met).

Each segment of intersecting lines is inspected for both criteria. In order to check criterion 1, the shortest distance to the boundary of the lake is computed for both end vertices and for the middle point of the segment. Only the minimum distance for the end vertices is kept and compared to the middle point distance. If the middle point distance is significantly longer (a factor of 1.1 was used in the tests), the criterion is met (Figure 8a and b). In order to check criterion 2, the orientation of the tested intersecting segment is computed and compared to the mean of the orientations of the shore segments around the tested segment. If the angle difference is close to  $90^\circ$  (in fact bigger than  $60^\circ$ ), the criterion is met (Figure 8c and d). When both criteria are met, the segment is considered as part of a bridge. Figure 7d shows that the identification of bridges greatly improves the automatic harmonization.

## Bus Stop Displacement

Bus stop displacement is an instance of logical displacement of bus stops that cannot be located where they are captured. This has two possible causes: Either the bus stop is misplaced by lack of precision, or the road that serves the bus stop is missing. The proposed algorithm has three steps: (i) Verify the context to decide if displacement is required; (ii) find the most probable serving road; (iii) compute the displacement vector for the most probable serving road.

The main criteria to decide whether displacement is required or not is the distance to roads: If it is over 12 m, the bus stop is considered misplaced unless a public transport area is also defined. If the bus stop is inside a building or other features, like sports fields, it strengthens the probability of a misplaced bus stop. In the case of Figure 9, the ambiguous bus stop is in the middle of a large building, which is unlikely a bus station (i.e. the only case where a bus stop should



be in a building) considering the semantics and the shape of the building. So the bus stop is most likely misplaced.

The most probable serving road is first computed by finding the nearest roads within a specified radius (roads further than 120 m are not considered in our experiments). The nearest road is the default serving road, but obstacles can discard this choice. The segment between the bus stop and its projection on the road is buffered and intersections with buildings, hydrological features or barriers are searched. If one obstacle is found, the nearest other road without an obstacle is then considered as the most probable serving road. But if the bus stop already intersects a building (like in Figure 9), this building is not considered as an obstacle.



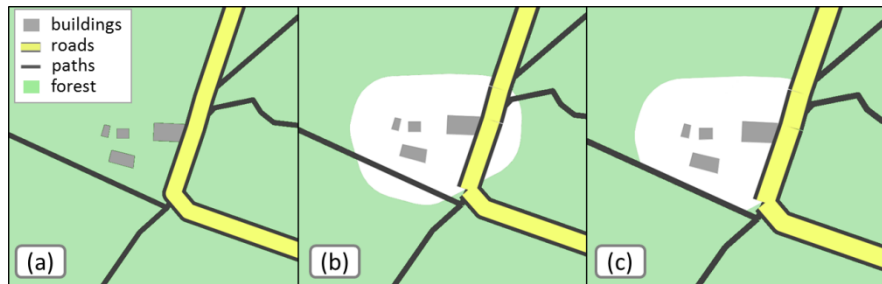
Fig. 9. One bus stop is far from the road and harmonized by displacement.

Finally, the displacement is computed in the direction of the most probable serving road, to be close to the road (the length of the road is estimated according to the semantics attached to the road), and in a location with space. This location with space is computed the same way as in Duchêne et al. (2012), by removing the spaces already occupied by other objects (the same objects used to find obstacles). Figure 9 shows an example of a bus stop automatically displaced.

## *Clearing Addition*

### **Algorithm Description**

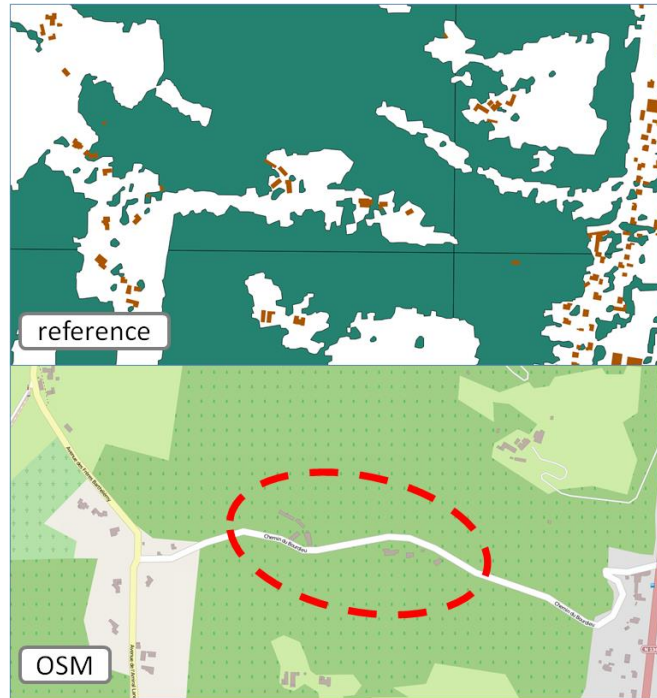
In the case of buildings grouped inside a forest, the proposed algorithm computes the extent of the building group and then computes a probable clearing geometry from the group extent and the surrounding objects. First, buildings are grouped using a process similar to Boffet (2000) and Chaudhry and Mackaness (2008), and similar to the built-up extension algorithm, merging buffers computed around each building inside a forest. The buffer used here is larger, 25 m, in order to create a larger group of buildings. However, the choice of a unique buffer parameter value, effective on a large area, is a complex problem. See section 5.1 for further discussions. Then, when groups are identified, the buffer of the convex hull of the group is computed, as a gap between buildings and trees is quite probable (Figure 10b). Finally, the network helps to refine the clearing extent (Figure 10c). The clearing polygon is cut into several parts using intersecting roads and paths, and the small parts that contain no building are removed from the clearing geometry. As roads and paths are often “natural” boundaries to clearings/forests, this last step results in a more realistic clearing boundary.



**Fig. 10.** a) an initial building group inside a forest, b) a first clearing geometry is computed by merging the dilated buildings, c) removal of clearing parts that lie over a path or a road.

## Comparative Evaluation

Reference data from IGN, the French national mapping agency, were used to compare some harmonized results to existing high quality maps. Comparing clearings automatically obtained by this algorithm to reference clearings is a complex task because the reference is quite different from the OSM forest data (Figure 11). In particular, the difference is illustrated by clearings in the harmonized map that are not clearings in the reference, but just a recess in the forest (Figure 11).



**Fig. 11.** Obstacles to conduct comparative evaluation: The clearing is not a clearing in the reference data, but only a recess in the forest.

In order to compare clearings as two polygons, we have to close the clearings in the reference data when they actually are not holes. This is done by intersecting reference forests with an expanded envelope of the harmonized clearings. Then, shapes of clearings are compared using the surface distance between two polygons (Girres and Touya 2010) described in equation 1. Comparisons were carried out on 33 clearings automatically created in three different areas in France. The average surface distance is 0.73, which is quite a large value as 1 stands for disjoint polygons and 0 for equal polygons. However, we consider this as a validation that harmonized clearings approximately occupy the same space as actual clearings. The aim never was to create more realistic clearings, which is impossible only using OSM buildings.

$$surface\_distance(A, B) = 1 - \frac{area(A \cap B)}{area(A \cup B)} \quad (1)$$



**Fig. 12.** Comparison of harmonized and reference clearings. a) and b) the actual clearings are bigger than the harmonized ones. c) The lack of ponds (to be included in the clearing) in the OSM makes shapes quite different.

Figure 12 shows some clearing harmonizations used in this comparative evaluation with the reference forest data. Figure 12c shows that other objects (in this example, ponds) should be included in the computation of clearings and also that the incompleteness of OSM greatly penalizes harmonization, since ponds displayed in Figure 12c are not captured in OSM, but are extracted from the reference data.

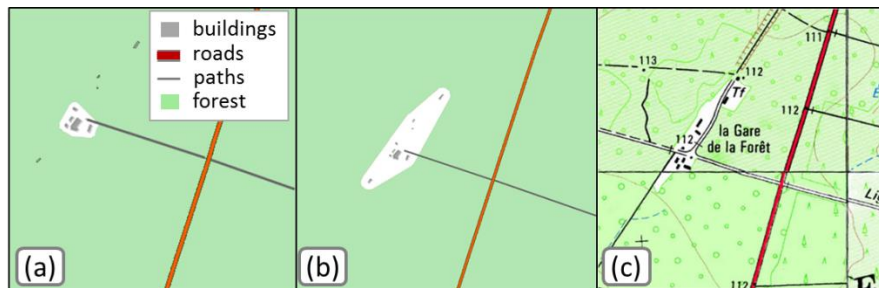
## Discussion

### *The Importance of Parameterization*

Experiments on large datasets from different countries and landscapes confirmed our assumption on the difficulty to find the best parameter values for the proposed harmonization algorithms. First, harmonization algorithms are hard to parameterize as parameter values are hard to correlate with visual results. It is a classical problem in automatic mapping processes, such as map generalization algorithms (Weibel et al. 1995), or label placement. For instance, there is no obvious value for defining how far a building can be considered to be “just outside” a built-up area.

Moreover, it appears that harmonization algorithms parameters are context-dependent insofar as parameter values are adapted to some situations and other situations require different parameter values. For instance, Figure 13a clearing uses the standard set of parameters empirically defined (25 m buffer around buildings to cluster the buildings), but does not look like the real clearing drawn in the IGN map (Figure 13c). A specific set of parameters (75 m buffer radius to cluster the buildings), which fails for most other cases, gives, here, much

better results. This example suggests that trying different set of parameters and keeping the best result, and/or defining situation-specific parameters, might be a more robust solution than finding the best parameters and applying them everywhere.



**Fig. 13.** The clearing created with standard parameters (a) does not look like the clearing in the IGN map (c); however, different parameter values give a closer result to the IGN map (b), but would fail in most other cases.

### ***Required Cartographic Knowledge and Degree of Automation***

The automation of cartographic processes, such as label placement, style definition, or generalization often requires some acquisition and formalization of the knowledge of cartographers (Buttenfield and McMaster 1991). Automatic harmonization of large scale OSM maps does not avoid this bottleneck, and some kind of cartographic knowledge base is necessary for several steps of the process.

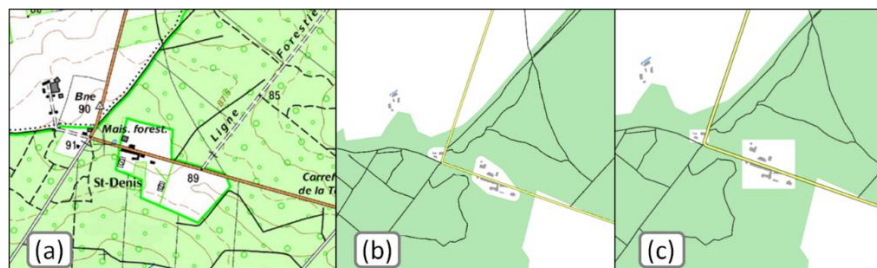
The first type of cartographic knowledge to formalize to enable harmonization is the identification of the key anomalous spatial relations. Some have been identified in this paper, but the list is not exhaustive, and additional relations may be of interest in other parts of the world (only French OSM datasets are used in this work) where landscapes are different and the detailed and rough features may not be the same. Then, there is what Taillandier, Duchêne, and Drogoul (2011) call *control knowledge*, which allows the definition of good parameters for automatic operations, and also allows the guidance of processes that chain several operations. Control knowledge requires experimenting with harmonization techniques to find out what leads to better maps.

In order to acquire this knowledge base in a more generic way, learning and artificial intelligence techniques could be used (Weibel et al. 1995). Formalizing the knowledge to share it would also be beneficial, and the collective project of building an ontology for on-demand mapping processes (Gould et al. 2014) could be used in this way.

Then, this knowledge could be used in processes able to chain harmonization operations, to adapt parameters to specific geographical situations, and to handle propagations of transformations. It could be interesting to adapt optimization (e.g., Harrie and Sarjakoski 2002, Sester 2005) or multi-agent techniques (e.g. Duchêne et al. 2012) used in automated map generalization to enable such harmonization processes. Nevertheless, problems should be less complex than map generalization, as large scale map symbols allow more free space in the map than small scale map symbols.

### ***Abstraction versus Realism***

Harmonization operations try to guess what the consistent detailed information would be from the detailed objects of the dataset. Thus, information that does not correspond to ground truth is introduced into the map, in order to make the map readable. For the same reason, map generalization also distorts ground truth, by moving or simplifying objects. Thus, it is necessary to wonder if harmonization should aim at realistic harmonized representations, which mislead the reader and making him/her believe that the map is a realistic view of ground truth. Or, aim at abstract harmonized representations that show to the map reader that the information is not exactly as it is represented on the map. Figure 14 illustrates both strategies for the clearing creation around building groups. None of the representations is close to ground truth, or even to its representation in the IGN map. But Figure 14b is clearly a more realistic representation of the clearing than Figure 14c, which is more abstract or sketchier.



**Fig. 14.** (a) The actual shape of a clearing in the IGN map; (b) a computed realistic shape; (c) a computed abstract/schematic shape

Research in computer graphics and non-photorealistic rendering (like maps) show that blurring an object or making it sketchier may convey information on data quality (Wood et al. 2012). Dashes also proved to convey uncertainty information (Boukhelifa et al. 2012). So, a sketchy or dashed clearing outline could also be an efficient alternative to our proposed realistic or schematic harmonization. In the case of bus stop disambiguation, some instances may remain

unsolved (Figure 15), and blurring or sketching the bus stop symbol could convey the uncertainty and avoid misinterpretation for the map reader.



**Fig. 15.** Unsolved disambiguation (which road serves the stops?) could be overcome by blurring symbols to convey uncertainty

### *Update OpenStreetMap?*

In a certain way, harmonization operations improve OSM data quality and correct some mistakes in the database. So it may be tempting to use the harmonized operations to push updates in the OSM database. However, harmonization is a process dedicated to cartography, so some transformations carried out may be irrelevant for the OSM database. Operations that include some kind of caricature should not be used to improve the OSM database. For instance, the alignment of trees along roads caricatures the tree alignment to make it straight, in order to improve the map clarity. But positional accuracy is then lost for some trees. However, occurrences of trees badly aligned are consistency problems of the OSM database that should be corrected, but with different operations that focus more on placement accuracy than map legibility.

Some other operations could be pushed in the database as it improves the level of detail of some rough objects, but it should be done carefully as it breaks a general rule of OSM to rely on ground truth to contribute to the project. For instance, the extension of built-up area to nearby buildings improves the level of detail of the built-up area but there is no checking in the field or with images that the new extent is close to ground truth.

Finally, we believe that some operations could be included without further checking, because the modifications are sure. For instance, “Adjust shape” operations avoid situations that cannot exist, like a path intersecting a lake or a



forest in the sea. Existing OSM tools like KeepRight<sup>1</sup> or Osmose<sup>2</sup> already search for such kind of problems, for further manual corrections by OSM contributors.

## Conclusion

To conclude, this paper tackles a new cartography problem raised by OpenStreetMap data, namely the level of detail harmonization that improves the level of detail of some rough objects in large scale maps. Several types of harmonization operations are proposed and experimented on OSM datasets. Some problems raised by the experiments are discussed, such as the need for realism or abstraction and the automation of the proposed algorithms in a process to harmonize a complete map. The first results show that it is a promising topic to explore in automated cartography.

Further research should clearly focus on harmonization processes, to be able to automatically chain harmonization operations, and solve complex problems that involve many objects and require optimization techniques. The processes should tackle the dependency of parameters on the geographic context of features. Of course, each operation presented in the paper could be improved and new operations have to be designed for the LoD inconsistencies that are not mentioned in the paper. Furthermore, as harmonization operations transform data into something realistic but false, abstract harmonization should be investigated with user tests, to know if map user better understand realistic or abstract harmonization. Finally, one of the main characteristics of OSM is that contribution patterns change all over the world (Jokar Arsanjani et al. 2015) and the inconsistencies encountered in the datasets tested in this paper might not occur elsewhere. Recent research showed that there is some relation between social aspects and contribution patterns in a region of the world (Mashhadi, Quattrone, and Capra 2015), so it can be inferred that LoD might differ according to these varying patterns. As a consequence, experiments should be carried out to analyze the influence of varying contribution patterns on LoD harmonization problems.

## References

- Biljecki F, Ledoux H, Stoter J, Zhao J (2014) Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems* 48:1–15
- Boffet A (2000) Creating urban information for cartographic generalisation. In: *International Symposium on Spatial Data Handling*, Beijing, China, 10–12 August 2000

---

<sup>1</sup> <http://keepright.ipax.at/>

<sup>2</sup> <http://osmose.openstreetmap.fr/fr/map/>



- Boukhelifa N, Fekete JD, Bezerianos A, Isenberg T (2012) Evaluating sketchiness as a visual variable for the depiction of qualitative uncertainty. *IEEE Transactions on Visualization and Computer Graphics* 18(12):2769–2778.
- Bucher B, Brasebin M, Buard E, Grosso E, Mustière S, Perret J (2012) GeOxygene: Built on top of the expertness of the french NMA to host and share advanced GI science research results. In: Bocher E, Neteler M (eds) *Geospatial Free and Open Source Software in the 21st Century*. Springer, Berlin, Heidelberg, p 21–33
- Burghardt D, Petzold I, Bobzien M (2010) Relation Modelling within Multiple Representation Databases and Generalisation Services. *The Cartographic Journal* 47(3):238–249
- Burghardt D, Duchêne C, Mackaness WA (eds) (2014) *Abstracting Geographic Information in a Data Rich World: Methodologies and Applications of Map Generalisation*. Springer, Heidelberg
- Buttenfield B, McMaster R (eds) (1991) *Map Generalization. Making Rules for Knowledge Representation*. Longman, London
- Chaudhry OZ, Mackaness WA (2008) Automatic identification of urban settlement boundaries for multiple representation databases. *Computers, Environment and Urban Systems* 32(2):95–109
- Chaudhry OZ, Mackaness WA, Regnaud N (2009) A functional perspective on map generalisation. *Computers, Environment and Urban Systems* 33(5):349–362
- Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10(2):112–122
- Duchêne C, Ruas A, Cambier C (2012) The CartACom model: transforming cartographic features into communicating agents for cartographic generalization. *International Journal of Geographical Information Science* 26(9):1533–1562
- Foerster T, Stoter J, Köbben B (2007) Towards a formal classification of generalization operators. In: *Proceedings of 23rd International Cartographic Conference*. Moscow, Russia, 2007
- Girres JF, Touya G (2010) Quality assessment of the french OpenStreetMap dataset. *Transactions in GIS* 14(4):435–459
- Gould NM, Mackaness WA, Touya G, Hart G (2014) Collaboration on an ontology for generalisation. In: *Proceedings of 17th ICA Workshop on Generalisation and Multiple Representation*. Vienna, Austria, 2014
- Haklay M (2010) How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design* 37(4):682–703
- Harrie LE, Sarjakoski T (2002) Simultaneous graphic generalization of vector data sets. *Geoinformatica* 6(3):233–261
- Jokar Arsanjani J, Zipf A, Mooney P, Helbich M (2015) An introduction to OpenStreetMap in geographic information science: Experiences, research, and applications. In: Jokar Arsanjani J, Zipf A, Mooney P, Helbich M (eds) *OpenStreetMap in GIScience*. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, p 1–15
- Klammer R (2013) TileGen - an open source software for applying cartographic generalisation to tile-based mapping. In: *Proceedings of 26th International Cartographic Conference*. Dresden, Germany, 2013
- McMaster RB, Shea KS (1988) Cartographic generalization in digital environment: A framework for implementation in a GIS. In: *Proceedings of GIS/LIS'88*, San Antonio, Texas, USA, 1988, p 240–249
- Mackaness WA (2007) Understanding geographic space. In: Mackaness WA, Ruas A, Sarjakoski T (eds) *The Generalisation of Geographic Information: Models and Applications*. Elsevier, Amsterdam, p 1–10
- Mackaness WA, Chaudhry OZ (2011) Automatic classification of retail spaces from a large scale topographic data-base. *Transactions in GIS* 15(3):291–307

- Mackaness WA, Burghardt D, Duchêne C (2014) Map generalisation: Fundamental to the modelling and understanding of geographic space." In: Burghardt D, Duchêne C, Mackaness WA (eds) *Abstracting Geographic Information in a Data Rich World*. Springer, Heidelberg, p 1–15
- Mashhadi A, Quattrone G, Capra L (2015) The impact of society on volunteered geographic information: The case of OpenStreetMap. In: Jokar Arsanjani J, Zipf A, Mooney P, Helbich M (eds) *OpenStreetMap in GIScience*. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, p 125–141
- Monmonier M (1996) *How to Lie with Maps* (2nd Edition). University Of Chicago Press, Chicago
- Mooney P, Corcoran P (2012) The annotation process in OpenStreetMap. *Transactions in GIS* 16(4):561–579
- Mustière S, Moulin B (2002) What Is Spatial Context in Cartographic Generalisation? In: *Proceedings of Joint International Symposium and Exhibition on Geospatial Theory, Processing and Applications*, Ottawa, Canada, 2002, p 274–278
- Reimer A, Kempf C, Rylov M, Neis P (2014) Assigning scale equivalencies to OpenStreetMap polygons. In: *Proceedings of AutoCarto 2014*, Pittsburgh, USA, 2014
- Regnauld N, McMaster RC (2007) A synoptic view of generalisation operators. In: Mackaness WA, Ruas A, Sarjakoski LT (eds) *Generalisation of Geographic Information*. Elsevier, Amsterdam, p 37–66
- Roth RE, Brewer CA, Stryker MS (2011) A typology of operators for maintaining legible map designs at multiple scales. *Cartographic Perspectives* 68:29–64
- Salichtchev KA (1983) Cartographic communication: A theoretical survey. In: Taylor DRF (ed) *Graphic Communication and Design in Contemporary Cartography*, Volume 2. Wiley, New-York, p 11–36
- Schmid F, Janetzek H (2013) A method for high-level road network extraction of OpenStreetMap data. In: *Proceedings of the International Cartographic Conference 2013 (ICC 2013)*, Dresden, Germany, 2013
- Sester M (2005) Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science* 19(8): 871–897
- Sester M, Jokar Arsanjani J, Klammer R, Burghardt D, Haunert JH (2014) Integrating and Generalising Volunteered Geographic Information. In: Burghardt D, Duchêne C, Mackaness WA (eds) *Abstracting Geographic Information in a Data Rich World*. Springer, Heidelberg, p 119–155
- Smith B, Varzi AC (2000) Fiat and bona fide boundaries. *Philosophy and Phenomenological Research* 60(2):401–420
- Stoter J, Burghardt D, Duchêne C, Baella B, Bakker N, Blok C, Pla M, Regnauld N, Touya G, Schmid S (2009) Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems* 33(5):311–324
- Taillandier P, Duchêne C, Drogoul A (2011) Automatic revision of the control knowledge used by trial and error methods: Application to cartographic generalisation. *Applied Soft Computing* 11(2):2818–2832
- Touya G (2012a) What is the level of detail of OpenStreetMap? In: *Workshop on Role of Volunteered Geographic Information in Advancing Science: Quality and Credibility*, Columbus, Ohio, USA, 2012
- Touya G (2012b) Social welfare to assess the global legibility of a generalized map. In: Xiao N, Kwan MP, Goodchild MF, Shekhar S (eds) *Geographic Information Science 7th International Conference, GIScience 2012*. Springer, Berlin, Heidelberg, p 198–211
- Touya G, Brando-Escobar C (2013) Detecting Level-of-Detail inconsistencies in volunteered geographic information data sets. *Cartographica* 48(2):34–143
- Touya G, Reimer A (2015) Inferring the scale of OpenStreetMap features. In: Jokar Arsanjani J, Zipf A, Mooney P, Helbich M (eds) *OpenStreetMap in GIScience*. Springer International Publishing, Switzerland, p 81–99

- Walter V (2008) Automatic interpretation of vector databases with a Raster-Based algorithm. In: Proceedings of ISPRS Congress XXXVII, Commission II, WG II/4, Beijing, China, 2008
- Weibel R, Keller S, Reichenbacher T (1995) Overcoming the knowledge acquisition bottleneck in map generalization: The role of interactive systems and computational intelligence. In: Frank AU, Kuhn W (eds) *Spatial Information Theory A Theoretical Basis for GIS*. Springer, Berlin, Heidelberg, p 139–156
- Wood J, Isenberg P, Isenberg T, Dykes J, Boukhelifa N, Slingsby (2012) Sketchy rendering for information visualization. *IEEE Transactions on Visualization and Computer Graphics* 18(12):2749–2758