Automatic Structure Detection and Generalization of Railway Networks

Sandro Savino, Guillaume Touya

Automatic Structure Detection and Generalization of Railway Networks

Sandro Savino*, Guillaume Touya**

* Department of Information Engineering, University of Padova, Italy
** COGIT – IGN France, 73 avenue de Paris 94165 Saint-Mandé France

Abstract. Unlike road or river networks, railway networks automatic generalization are missing to properly handle the detailed networks provided in current geo-datasets like OpenStreetMap. This paper proposes automatic methods to automatically identify key structures of railway networks, such as parallel main tracks, or fan and pack patterns inside large train stations. Then, algorithms based on the detected structures are proposed to generalize the railway networks. The algorithms are tested on real datasets, including OpenStreetMap data.

Keywords: railway network, generalization, structures, strokes, OpenStreetMap

1 Introduction

As long as they were simply integrated into geographical databases, railway networks were quite easy to include in automated mapping systems: the networks were not very dense and the shapes were mostly straight so map generalization was easy. But railway networks are more and more modeled in a realistic ways, capturing in datasets all the tracks of the networks, such as OpenStreetMap requires for instance (Touya & Girres 2014). In order to include these complex railway networks into small scale maps, specific map generalization algorithms are necessary. The automatic generalization of geographical networks (e.g. roads, rivers…) has been tackled many times due to the importance on such features in maps, and more generally, in geographical datasets (Thomson & Brooks 2007, Stanislawski et al. 2014). However, there has been very little focus on railway networks generalization (Touya & Girres 2014). Although some techniques extracted from previous work, like ‘strokes’ can be applied to the generali-

zation of railway networks, algorithms dedicated to their specific geographical structures need to be developed. This paper proposes such dedicated techniques to identify the main structures of railway networks, and to generalize the network and its structures.

Section 2 deals with the modeling of railway network structures and their automatic detection in datasets where each track is captured. Section 3 describes new algorithms to generalize railway networks. Experiments on several real datasets are presented in section 4. Finally, section 5 draws some conclusions and proposes further research.

2 Modeling Railway Networks

2.1 Modeling Railway Networks as Complex Objects

Inspecting railway networks quickly shows that they are composed of two very different parts: there are mostly tracks where trains run, composed by one or more lines running parallel, and areas where trains stop, as a train yard or a train station, composed by a big number of tracks crossing, diverging and stopping (Figure 1). Generalization-wise, this distinction is not that sharp: main tracks are not very different from small train stations as they both are composed of parallel lines; big stations instead, are very different, because of the very big number of tracks, patterns and structures in them, requiring to use different techniques be generalized.

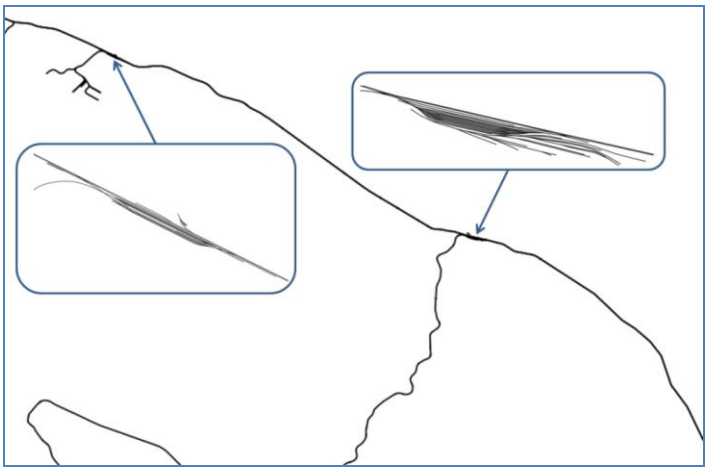

Figure 1. Railway network extracted from OpenStreetMap: it is mainly composed of main tracks with spare sidetrack areas.

In this paper, we will split the railroad network in main tracks and sidetrack areas: the first object corresponds to small groups of parallel tracks, pre-

sent in train lines and small stations while the latter corresponds to big groups of tracks, present in stations or train yards. Each object is separately modeled as it will require specific generalization; a method to automatically differentiate main tracks from sidetracks is proposed in Section 2.2.

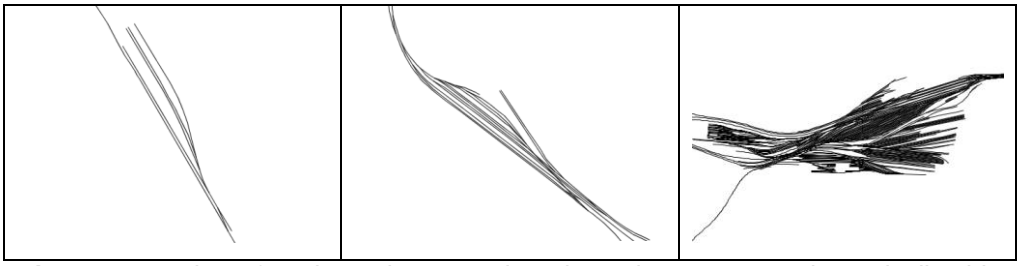

Figure 2. Number of tracks and patterns in train stations can vary dramatically: this requires a flexible algorithm able to cope with the different complexity.

When multiple parallel railway tracks are abstracted into a single railway route in most classical geographic datasets, highly detailed railway networks contain each track represented as a line. The standard distance between two parallel tracks is less than 10 meters, which does not allow the display of all these parallel tracks in most cartographic representations.

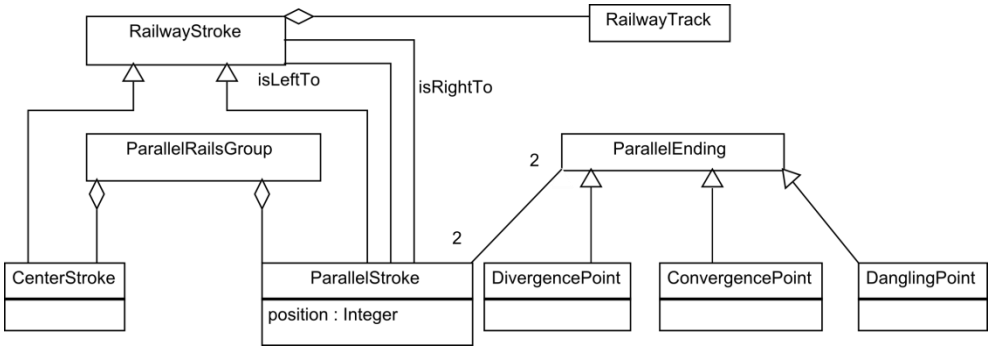

Figure 3. UML class diagram of railway main tracks modeled as complex objects.

In order to abstract parallel tracks as a single railway route, we propose to model railway main tracks into *Parallel Rails Groups* (Figure 3). This model is firstly based on the computation of strokes, *i.e.* perceptual complex entities that gather tracks that follow each other like a pen stroke (Thomson & Richardson 1999). Strokes are commonly used to identify salient routes in geographical networks (Thomson & Brooks 2007). A *Parallel Rails Group* is composed of a center stroke to which parallel smaller strokes are aggregated. The way parallelism ends between a parallel stroke and its central stroke is also identified with three types of endings: converging points (i.e. points where two parallel tracks meet), diverging points (i.e. points where two parallel tracks follow different directions), and dangling points (Figure

4). A position number is associated to each parallel stroke: 1 means that the stroke is directly on the right of the center stroke, 2 means that the stroke is on the right of a parallel stroke with a position of 1, -1 means that the stroke is on the left of the center stroke, etc.

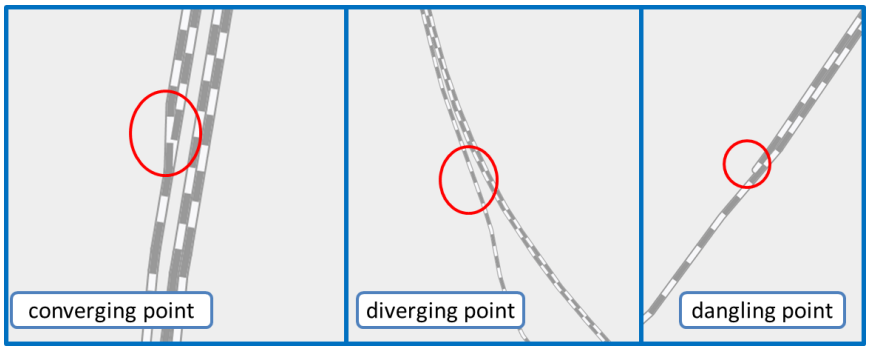

Figure 4. Three types of parallelism endings inside parallel railway groups.

The structure of the network changes dramatically in the surrounding of a train yard or a train station: the tracks composing the main line split in different directions and are sided by multiple parallel tracks (see zoomed areas in Figure 1); tracks can converge, diverge, cross each other and also stop in a dead end. We call such a group of tracks a sidetrack group.

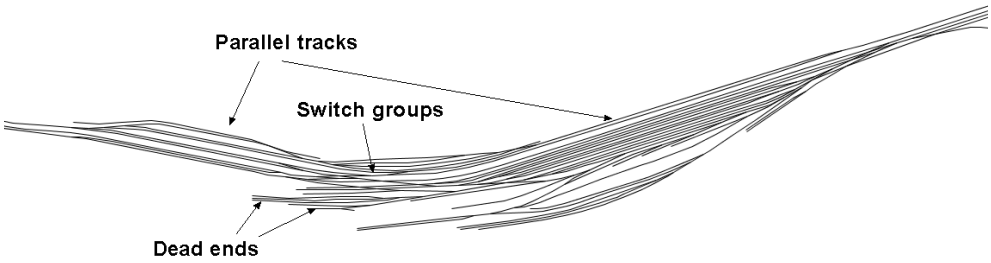

Figure 5. Structures that can be recognized in a train station.

It's important to note that main tracks are interrupted by the presence of a sidetrack group: due to the complexity of the line work inside these structures, it is often not possible to identify a single track or stroke entering and exiting the sidetrack group. Different characteristic structures can be identified in a sidetrack group (Figure 5):

- tracks diverging from a group of switches, running parallel and then converging in another group of switches,
- dead ends, where multiple parallel tracks are used to store freight cars,
- switch groups, where many tracks meet and cross each other.

Sidetrack groups capture important feature of the rail network (e.g. a logistic center, a major train station) and represent a characteristic that should be preserved in the generalized data; their large number of tracks (up to hundreds) makes them a prominent feature of the network but at the same time makes them very hard to generalize. In order to succeed, it is necessary to reduce the complexity of the generalization task by identifying some objects into which to decompose each group.

The objects described before are well defined, independent one from each other and they are present, in different size and numbers, in every big station. In our strategy we decided then to model sidetrack groups as composed by packs, fans and free tracks. The UML class diagram in Figure 6 shows the relation between these components, while section 2.3 will describe further each of them.

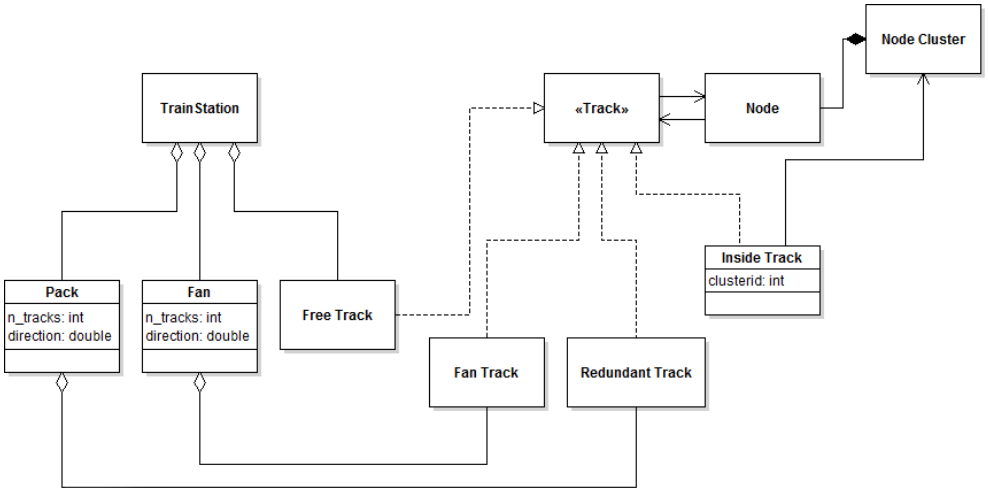

Figure 6. UML Class diagram of the sidetrack group model and its components.

2.2 Automatic Detection of Main Tracks and Sidetracks

The most prominent difference between main tracks and sidetracks is the amount of tracks that compose each one. The differentiation technique implemented draws a line segment for each track, placed in the middle of it and orthogonal to its direction. The algorithm then counts how many tracks each segment crosses.

If the number of crosses is above a threshold, the track is candidate to be part of a sidetrack group. Candidates are then clustered by proximity: if a cluster of candidates is composed by only a few tracks, it does not qualify as a sidetrack group and its tracks are flagged as main tracks. The thresh-

olds were chosen experimentally and were set to 60 meters for the segment length, and 10 for the cluster group number. Once the group of tracks composing a sidetrack group has been identified, the main tracks connecting the rail network to the group are flagged as *in-out tracks*.

2.3 Automatic Structure Detection

2.3.1 Parallel Railway Groups

The first step of the detection process is to compute strokes within the main tracks. Then, strokes are ordered regarding their length, and the longest one is used as the first center stroke for a new parallel railway group. Then, the remaining strokes that are within a 10 meter proximity area (computed with a dilatation operator) for a significant distance (empirically set to 100 m) are considered as parallel strokes. Then, strokes parallel to each of the new parallel strokes are searched recursively. Once parallel railway groups have been identified, the types of parallelism endings are characterized for each parallel track, in order to ease the reconnections after the collapse of the parallel tracks (see section 3.1). When the parallel geometries intersect, it is a converging point, when they do not intersect but always stay within the minimum distance, it is a dangling point, and when the parallel geometry leaves the minimum distance area, it is a diverging point.

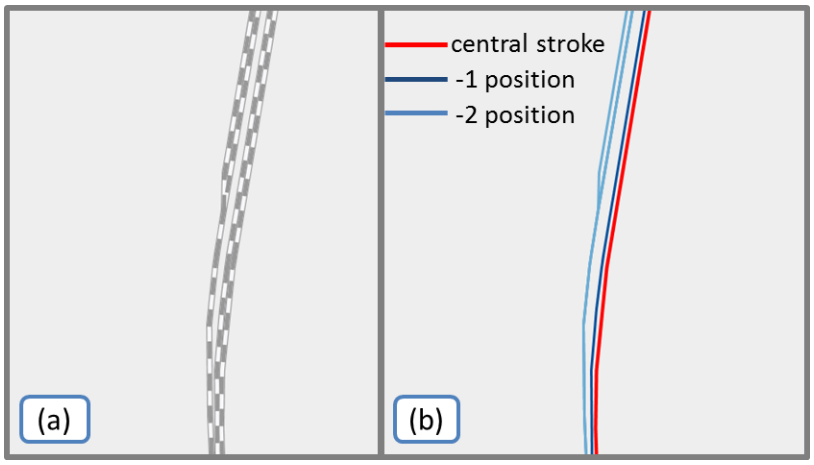

Figure 7. parallel railway tracks that are automatically groups around the longest stroke (in red).

A final step checks if the collapse altered the connections between the selected railway routes. Figure 8 shows a case where the connection between two railway routes is made only with the parallel track, and once this track is removed, the connection is lost. In such cases, the lost connections are restored by extending the lines to the nearest selected railway. The

parallel railway groups detection is not carried out on tracks that are part of a sidetrack group because such tracks are characterized, and then generalized differently (see next sections).

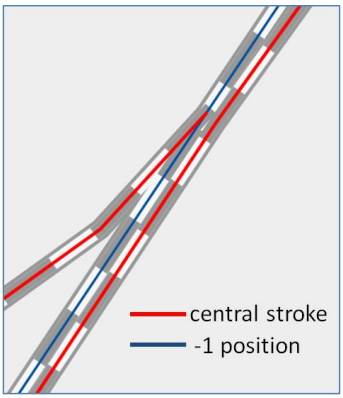

Figure 8. The connection between the railway lines is not between the center strokes so the removal of the parallel stroke alters the connection.

2.3.2 Detection of Structures inside Sidetrack Groups

The most important structures in a sidetrack group are packs and fans. Tracks belonging to a sidetrack group that are neither part of a fan nor of a pack, are classified as cluster tracks or free tracks.

Fans are basically formed by dead end tracks; what differs them from standard dangling edges is the fact that they are formed by groups of parallel, evenly spaced tracks that merge into each other (usually two by two), forming a comb or tree like structure with many branches. The detection algorithm developed is able to detect the whole tree structure up to the stem by walking the network starting from the dangling edges, and giving a score to each track visited. The score of each edge is a function of the score of the edges surrounding it; a dangling edge has a score of 100, his children have 50; if the value of an edge is 100, it becomes part of the fan and can propagate the selection to his children. At the end of the process, all edges having score of 100 are flagged as *fan tracks*.

Since fans are a “local” structure, a length threshold is used to stop the propagation on long tracks (the threshold was set to 100 meters). Once the process is finished, the tracks flagged as *fan track* are grouped into fans by clustering them on proximity (Figure 9).

Another very important structure in sidetrack groups are *packs*. Packs have some trait in common with fans, as they both are a group of parallel tracks stemming from the same area of the network, but they do not branch and, very important, they converge to another part of the network. The edges composing a pack are called r-edges, as redundant edges, because they

form a link between two areas of the network that are connected by many tracks. A pack is composed of all the tracks that connect the same two areas of the network. Generalization-wise, redundant edges are very important to detect as they are good candidates for selection since they can be eliminated without altering the connectivity of the network (Savino et al, 2010).

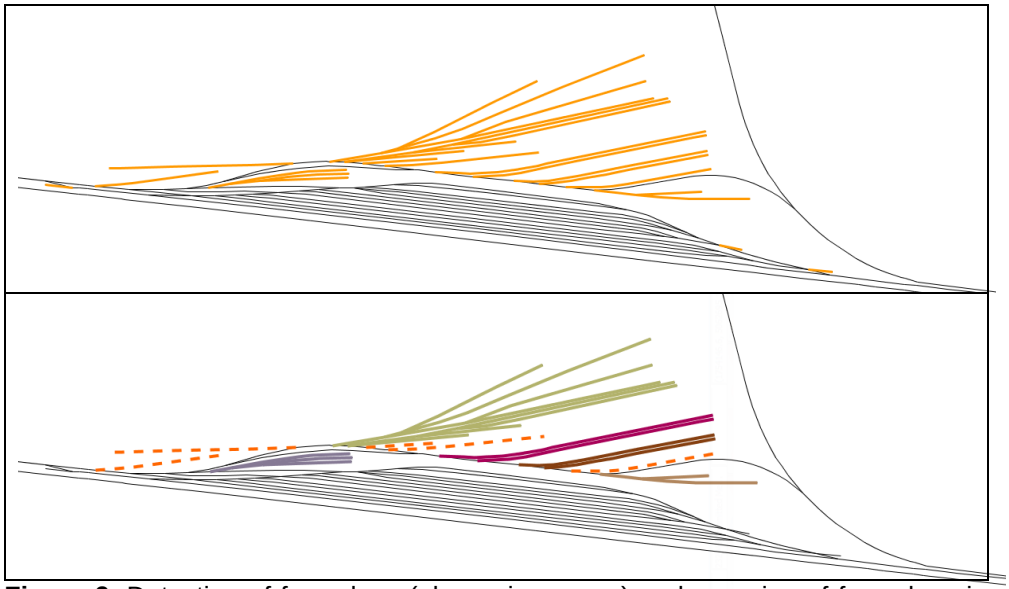

Figure 9. Detection of fan edges (above, in orange) and grouping of fan edges in fans (below, each color represent a different fan). Dashed lines represent fan tracks that have been filtered out and classified as single dangling edges.

To detect r-edges it is necessary first to cluster the nodes of the network: these clusters are useful to find those areas where the tracks of a pack converge. Clustering is performed on proximity and connection: two nodes are in the same cluster if they are closer than a threshold and they are directly connected by an edge. A unique identifier is assigned to each cluster and to each node that is not part of any cluster. The algorithm then calculates a connection code for each edge of the network: this code uniquely identifies the two nodes connected by each edge; the code is based on the identifier of each node or, if a node is part of a cluster, on the identifier of the cluster the node belongs to. Finally, the packs are created by grouping the edges by their connection code (Figure 10).The edges that connect two nodes that are part of the same cluster are flagged as *cluster tracks*: these tracks represent switch groups and play an important role in the topology of the network as they usually link packs connecting different clusters.

All the edges that are not part of fans or packs or are not cluster tracks, are classified as *free tracks*. These tracks cannot be further characterized: they have different direction and length and they roam “freely” between the other

structures defined above. These tracks also play an important role in a sidetrack group as they connect all the different structures together.

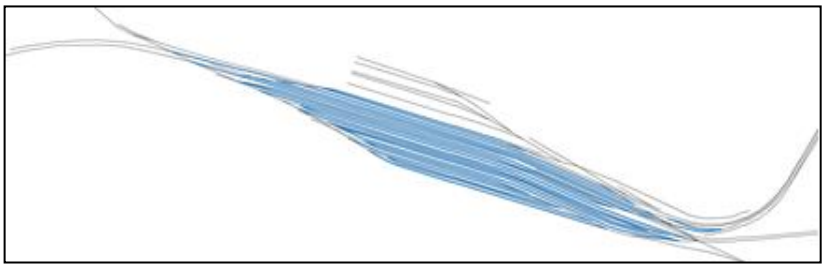

Figure 10. Detection of r-edges (in blue) and their grouping in packs (below, each color represents a different pack).

3 Algorithms to Generalize Railway Networks

3.1 Parallel Railway Tracks Collapse

Touya & Girres (2014) proposed an algorithm to collapse two parallel tracks into a center track that handles converging and diverging endpoints. So, when parallel tracks groups contain only two railway strokes, it can be used to collapse the group. However, Section 2 showed that parallel tracks groups are often composed of more than two parallel tracks, and the collapse algorithm cannot be directly extended to collapse more than two tracks. The proposed collapse algorithm is based on the parallel tracks group model described in the previous section. Only the railway tracks that belong to the center stroke are kept, and all parallel tracks are eliminated, while diverging tracks are reconnected to the center stroke (Figure 11).

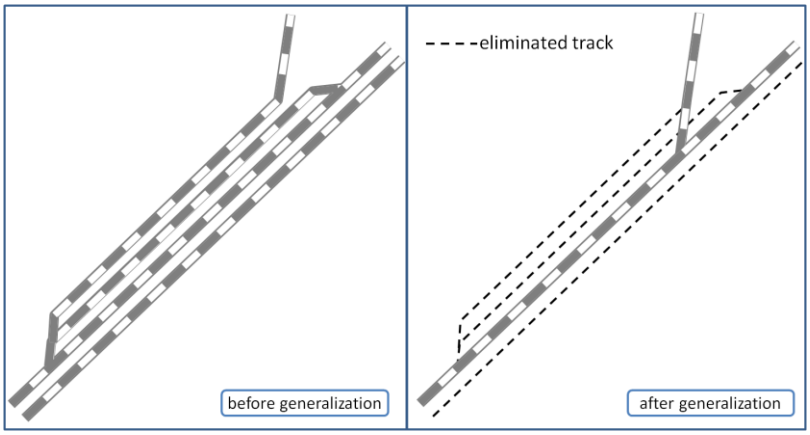

Figure 11. The collapse of parallel tracks group: only the center track is kept and diverging tracks are reconnected**.**

The diverging points of parallel tracks do not necessarily correspond to the extremity of a track feature geometry, so the geometry are cut to only keep the diverging part. Then, the first (or last) segment of the remaining geometry is extended to snap to the center stroke, which preserves the divergence angle, and avoids unnatural connections. Once parallel tracks have been collapsed, classical stroke-based selection algorithms can be applied to select only the important railway strokes (Thomson & Brooks 2007).

3.2 Sidetrack Groups Typification

Because of the different type of tracks that are present in a sidetrack group, it is difficult to find a generalization technique able to handle all the existing situations. In our approach we manage to generalize sidetrack groups by decomposing them in the previously described structures.

The strategy that we devised to generalize sidetrack groups is quite simple: each single structure is generalized independently and the final result is obtained by connecting every generalized structure to the network. As a design choice, the generalization is performed using only the selection operator; this limits the need to recreate the connectivity among the generalized features; on the other hand, it also poses some constraints on the shape of the resulting network. The first structures to be generalized are packs and fans. The most prominent characteristic of packs and fans is the parallelism of their tracks: the typification algorithm developed uses this characteristic to both select the tracks and maintain their initial pattern.

The algorithm calculates the direction of the group of tracks and then traces a line, orthogonal to the direction and centered on the centroid of the group; the intersection points between each track and this line are calculated: these points are used to measure the width of the group and calculate, on the base of the required minimum distance between tracks, which tracks should be selected (Figure 12). Analyzing the distance between the intersection points, the algorithm is able also to detect whether the tracks are regularly spaced or they can be clustered in sub-groups: in the latter case, selection is applied to every single sub-group.

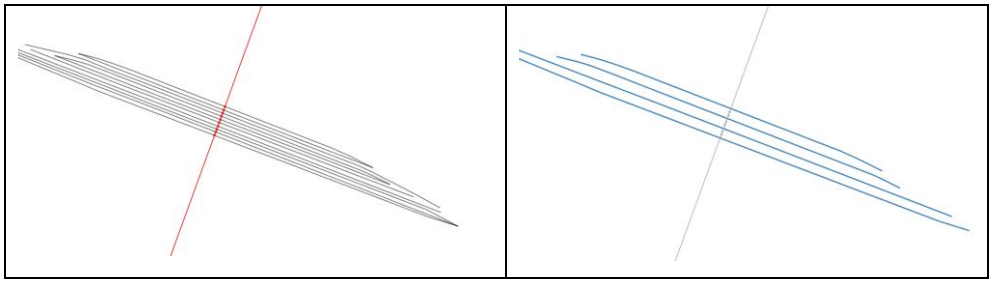

Figure 12. Pack generalization: a line orthogonal to the pack (left, in red) is used to characterize the pack and to select the tracks to keep (right, in blue).

This typification technique is used to generalize packs; because of the tree-like structure, though, it cannot be applied directly to fans: not all the fan edges have the same direction, and not all the fan edges would intersect the orthogonal line. For these reasons, the selection algorithm is applied only to the dangling edges of a fan; after the selection a special routine deletes all the fan tracks that are not needed anymore because they end up in a dangling edge that has been removed (Figure 13). Most of the packs are connected to cluster of nodes: after r-edges have been selected, cluster tracks are generalized accordingly, removing those tracks that are connected to redundant edges that have been deleted.

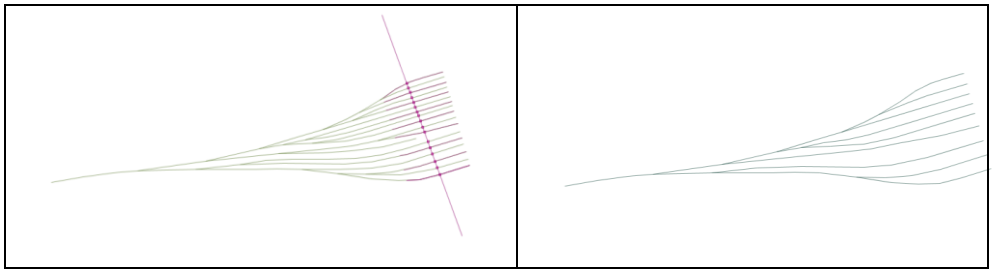

Figure 13. Fan generalization: same as packs, but applied to dangling edges (left, in purple); the edges connecting the deleted dangling edges are removed.

The final part of the process is the generalization of free tracks. Once they are separated from the other structures, free tracks, to some extent, resemble a main track and they can be both generalized using a similar stroke-based technique. To create the strokes, cluster tracks are added to the free tracks, in order to restore the connectivity in the areas where a stroke passes through a cluster. Furthermore, the in-out tracks (i.e. the edges of the main tracks connected to the sidetrack group) are also added to the set of edges that are used to create the strokes. The strokes are then generalized by detecting those too short and those too close to other strokes and either removing them or collapsing the shorter ones in the longer ones, with the collapse technique proposed for main tracks. Once the strokes have been generated and the free tracks generalized, the connection between the structures and the free tracks is restored; as a last step, cluster tracks that are not used are removed, together with single dangling edges that did not qualify as fan.

4 Experiments

Experiments have been carried out on two types of highly detailed railway networks: one depicting a whole French region (30,000 km²) was extracted from OpenStreetMap, and the other depicts an Italian region (18,000 km²) and was extracted from the regional database produced by the local gov-

ernment of the region of Venezia. Both datasets contain a big variety of data: small railroads composed by single lines, bigger lines composed by more tracks running parallel, small, medium and very big train stations.

Figure 14 shows some results of the main tracks collapse into one single track. The algorithm performs well on all types of parallelism endings and is able to quickly collapse large datasets.

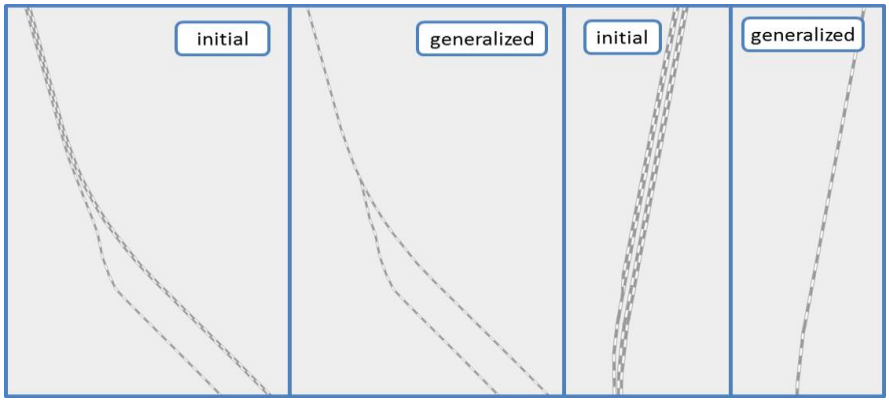

Figure 14. Two examples of parallel main tracks from OpenStreetMap collapsed into one with reconnections**.**

In Figure 15, Figure 16 and Figure 17 the results of our generalization algorithm applied to some stations of different complexity are shown. Packs are represented in blue, fans in yellow, free tracks in red, cluster edges in black, and the edges added to reconstruct the connectivity in green.

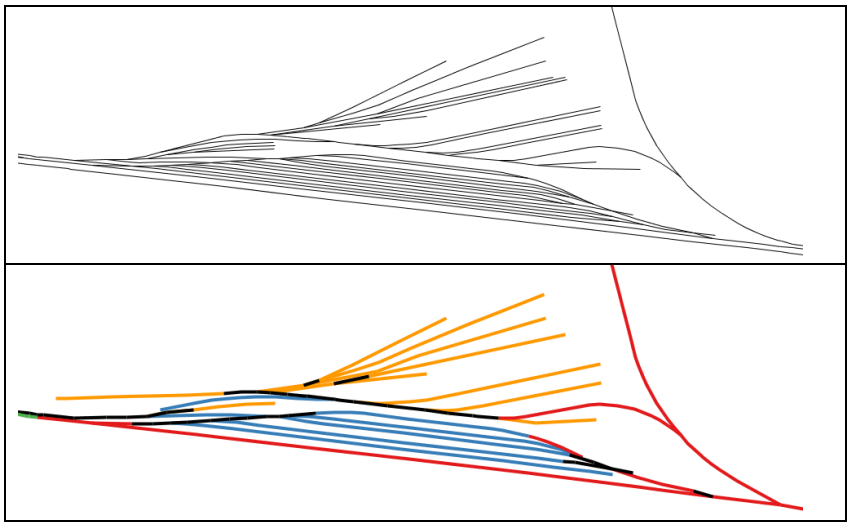

Figure 15. Generalization of a train station of medium complexity (input data above, generalized data below).

In general, the tests show that our algorithm performs well in detecting the different structures composing the network and, by dividing the problem in smaller generalization tasks, is able to handle also very complex stations. From the topological point of view, our algorithm guarantees that the generalized data is correct as no connectivity is lost due to the edge removal.

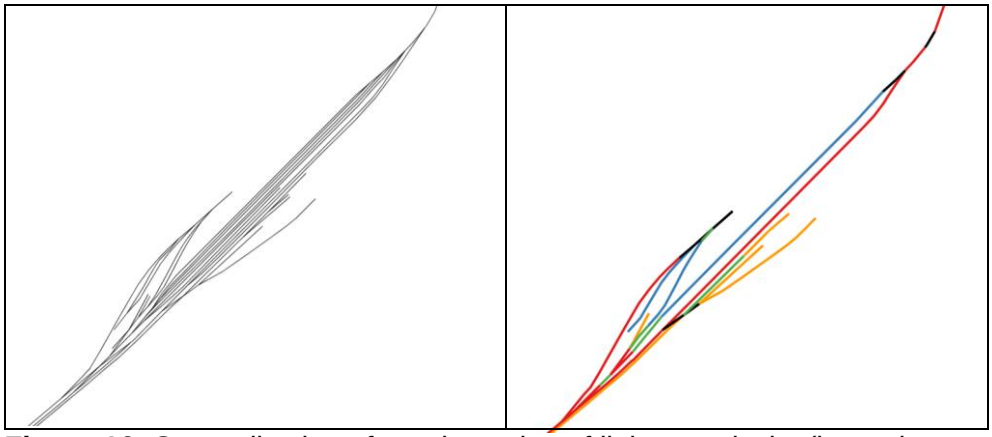

Figure 16. Generalization of a train station of little complexity (input data on the left, generalized data on the right)

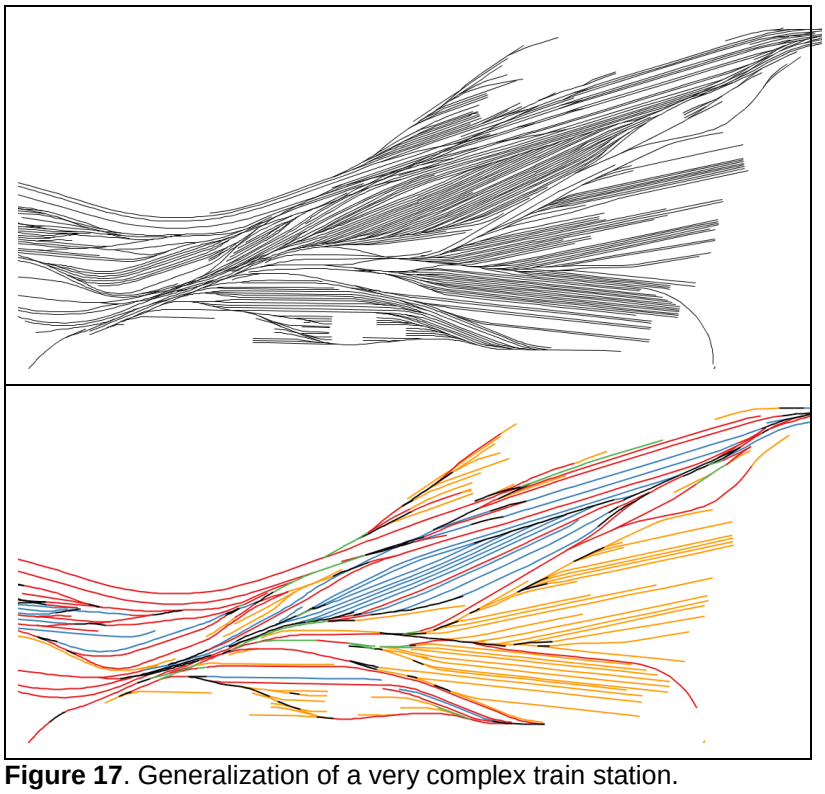

Figure 17. Generalization of a very complex train station.

The quality of the generalized data is in general good but leaves space to some improvements. In packs and fans, the use of selection as the main generalization operator, constrains the possible solutions for the typification of the tracks. Despite the ability of the algorithm to detect and handle patterns in the input, due to the layout of the original tracks the output can still present tracks that do not respect the minimum distance threshold. The typification technique adopted, however, makes it easy to compute displacement vectors that can be used to obtain a more uniform distribution of the generalized tracks.

In order to evaluate our proposals, the sidetracks generalization is compared to the algorithm proposed by Touya & Girres (2014). **Figure 18** shows some results on the big station in the Venezia area: the typification is good because the density decreased while the general pattern is preserved. However, the specific patterns of fans and packs are altered by this algorithm, with particularly some local loss of connections. This proves the need for an algorithm that handles such structures.

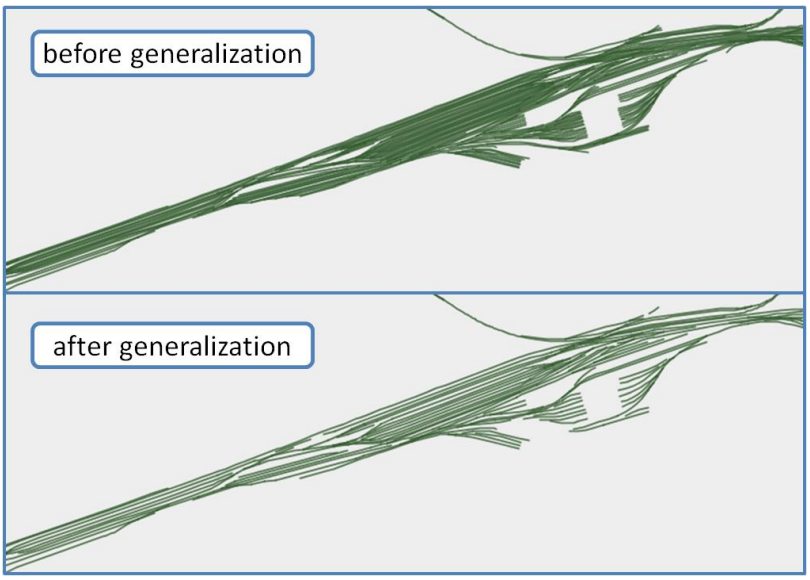

Figure 18. Results obtained on the Venezia area with the typification algorithm from (Touya & Girres 2014).

5 Conclusion

This paper presented an exhaustive approach for the generalization of large scale railroad networks. The input data is divided in two categories: parallel main tracks and sidetrack groups, i.e. train stations, composed by up to hundreds of tracks. The paper proposes for each of these objects a

model, identifying for each object its components or structures that can be detected using automatic spatial analysis techniques. The paper also proposes generalization algorithm based on the previously detected structures.

To go further, we believe that most of the proposed algorithm, for detection and generalization, can be improved. For instance, regarding the generalization of neighboring packs, since packs are generalized independently, there is no guarantee that the generalized edges will respect the minimum distance threshold across different packs. This is also true for free tracks near packs or running between two of them.Testing them on more datasets will help identifying particular cases where the algorithms do not perform as intended. There is also a need for a displacement/deformation operator to move the remaining tracks inside a station or a triage area apart (Bader at al. 2005). Such an operator would allow the reduction of the selection that can be too drastic.

References

Bader M, Barrault M, Weibel R (2005) Building displacement over a ductile truss. International Journal of Geographical Information Science 19(8):915–936

Grosso E, Perret J, Brasebin M (2012) GEOXYGENE: an interoperable platform for geographical application development. In: Bucher B, Le Ber F (Eds.), Innovative Software Development in Gis, pp. 67–90. John Wiley & Sons

Savino S., Rumor M., Zanon M., Lissandron I., 2010, Data enrichment for road generalization through analysis of morphology in the CARGEN project, 13th ICA Workshop, 2010, Zurich

Savino S, Rumor M, Zanon M (2011). Pattern recognition and typification of ditches. In: Ruas A (Ed.), Advances in Cartography and GIScience, pp. 425–437. Springer, Berlin, Heidelberg

Stanislawski LV, Buttenfield BP, Bereuter P, Savino S, Brewer CA (2014) Generalisation operators. In: Burghardt D, Duchêne C, Mackaness W (Eds.), Abstracting Geographic Information in a Data Rich World, Lecture Notes in Geoinformation and Cartography, pp. 157–195. Springer International Publishing, Berlin

Thomson RC, Brooks R (2007) Generalisation of geographic networks. In: Mackaness WA, Ruas A, Sarjakoski LT (Eds.), Generalisation of Geographic Information: Cartographic Modelling and Applications, pp. 255–267. Elsevier, Amsterdam.

Thomson RC, Richardson D (1999) The "good continuation" principle of perceptual organization applied to the generalization of road networks. In: Proceedings of 19th International Cartographic Conference

Touya G, Girres JF (2014) Generalising unusual map themes from OpenStreetMap. In: Proceedings of 17th ICA Workshop on Generalisation and Multiple Representation, Vienna, Austria