Progressive Annotation of Schematic Railway Maps

Yuka Yoshida*, Ken Maruyama*, Takamasa Kawagoe*,
Hsiang-Yun Wu†, Masatoshi Arikawa‡, and Shigeo Takahashi*,
* University of Aizu, Fukushima 965-8580, Japan
† TU Wien, A-1040 Vienna, Austria
‡ Akita University, Akita 010-8502, Japan

***Abstract*—Octilinear network layouts are commonly used as the schematic representation of railway maps due to their enhanced readability. However, it is often time-consuming to place station names on such railway maps by trial and error, especially within the limited labeling space around interchange stations. This paper presents a progressive approach to placing station names around stations in schematic railway maps for better automation of map labeling processes. The idea behind our approach is to annotate stations in dense downtown areas around the interchange stations first and then those in sparse rural areas. This is achieved by introducing the sum of geodesic distances over the railway network to identify the proper order in which to annotate stations. In the actual annotation process, we increase the labeling space around the railway network when necessary by progressively stretching railway line segments while retaining their original directions, which allows us to respect the original schematic layout as much as possible. We present several experimental results to demonstrate the effectiveness of the proposed approach, together with a discussion on parameter tuning in our formulation.**

***Index Terms*—Progressive annotation; geodesic distances; schematic layouts; railway maps; mixed-integer programming**

I. Introduction

Railway maps are a common tool for exploring a specific route over the transportation networks in major cities. Schematic layouts of such railway networks are often employed for better readability of topological connectivity among the railway lines. Among them, octilinear layouts are the most typical representation in which railway lines are aligned to horizontal, vertical, or 45-degree slanting directions. Several representative techniques [1]–[3] have been developed for automatically transforming geographic layouts of railway networks into their corresponding schematic maps.

However, annotating names of stations on such schematic railway maps usually incurs further technical problems because the central downtown area of a city map is usually congested with interchange stations together with multiple railway lines, when compared with sparse rural areas. This labeling problem becomes further complicated if we restrict the alignment of station names around the target stations to one of the eight octilinear directions. One possible solution is to manually create more labeling space in the central downtown area for placing annotation labels without overlap, which often results in a tedious trial and error process. Simultaneously, exploring the placement of all name labels is impractical due to the large search space arising from an uncountable combination of label alignment patterns.

This paper presents a progressive approach to automatically annotating station names on schematic railway maps. This is inspired by annotation guidelines proposed by Wu et al. in [4], in which the researchers started with the annotation of stations in the crowded downtown area including interchange stations and then gradually extended the annotated region to the sparse rural areas till reaching the dead ends of the railway lines. In our approach, we first sum up pairwise geodesic (i.e., shortest) distances for the respecting stations and sort them in an ascending order of the distance sum. This allows us to first annotate stations in the congested downtown area and then move on to those in rural areas. To align names with the octilinear directions, we adaptively elongate the railway segments while retaining their directions from the center of the map to its margin, to spare sufficient labeling space around the respective stations.

Figure 1 depicts how we can progressively annotate stations on a schematic railway map. First, we take an octilinear layout as input of a railway network (the top-left image of the figure). We then begin with annotating stations around the center of the network including interchange stations. We continue this annotation process step by step, while adaptively stretching the railway line segments if we lack the space for placing the station names. This process allows us to annotate all stations on the map while maximally retaining the overall layout of the railway network, as shown in the bottom-right image.

This paper is organized as follows: In Section II, we provide a brief survey of related studies. In Section III, we review the previous formulation for schematizing railway networks based on mixed-integer programming, which serves as a basis for our approach. In Section IV, we detail our approach to progressively annotating schematic railway maps. After having presented design examples together with a discussion on possible parameter tuning for our approach in Section V, we conclude this paper and refer to future work in Section VI.

II. Related Work

Various methods have been proposed for schematizing the geographic layout of railway network maps. In particular, constrained optimization techniques successfully align railway line segments to octilinear directions, i.e., horizontal, vertical, and 45-degree diagonal directions. A set of criteria for schematic railway maps are introduced and the formulation

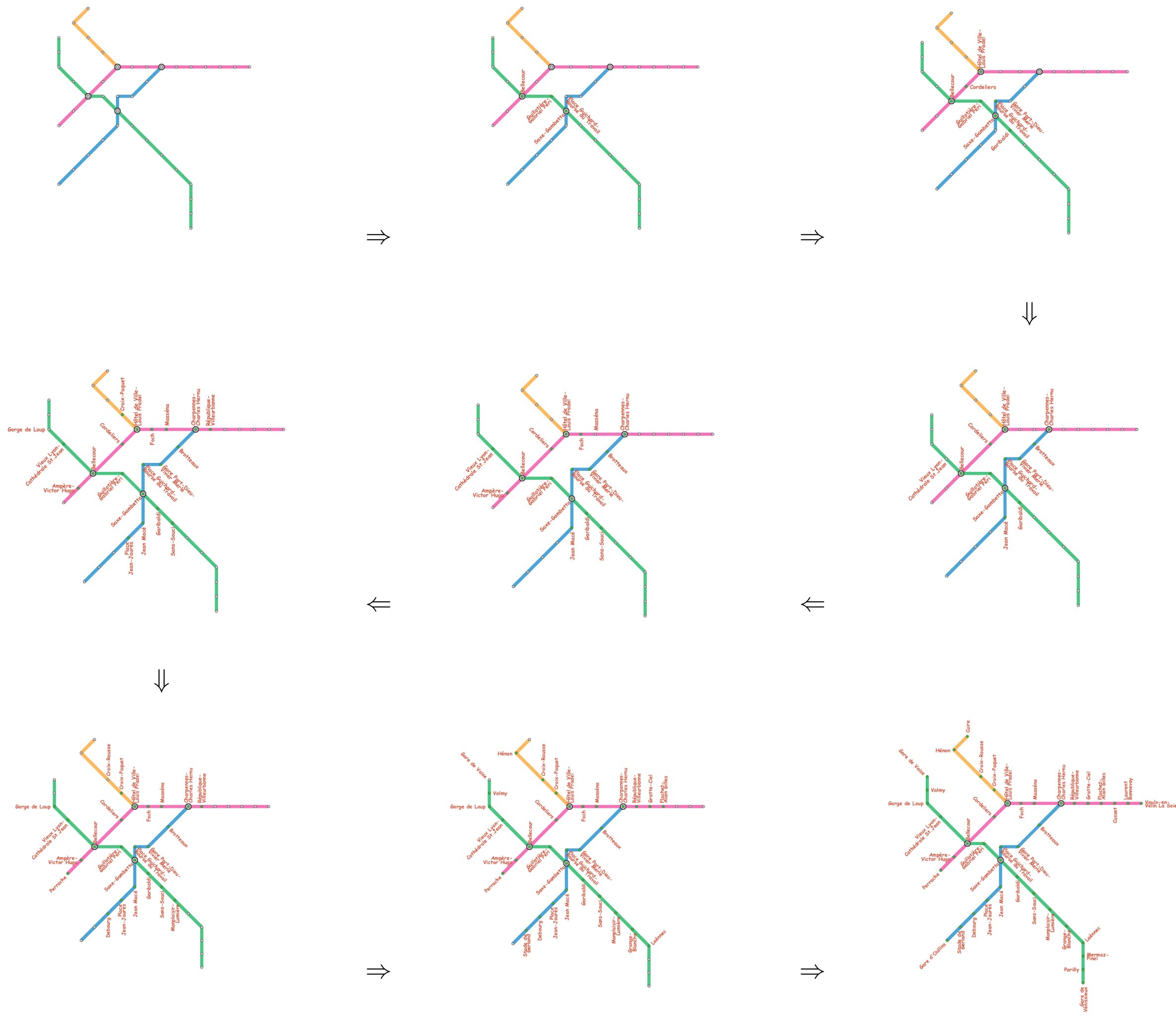


Fig. 1. Metro network in Ryon. Station names are placed from the map center while adaptively extending railway line segments for sparing labeling space.

is optimized using a hill climbing method [1] and a mixed-integer programming framework [2]. Wu et al. extended the work of [2] for elongating a specific line of interest as a central straight line [5] and a loop line [6] in the map. Another sophistication was achieved by Wang and Chi [3], who successfully accelerated the computation by formulating the optimization as a quadratic programming problem. Claudio and Yoon [7] developed a method of placing thumbnail images in areas of interest on railway maps, where the schematic representation is created through the distortion of the entire map. In this paper, we employ the technique of [2] to schematize the geographic layout of the target railway map as the foundation of our proposed approach.

In addition, placing annotation labels on maps is a primary research subject for cartographic and geospatial visualization. Labeling techniques can be roughly categorized into two types, i.e., internal labeling [8] for placing an annotation label in the neighborhood of each point feature on the map, and external labeling [9], [10] for placing a label away from the corresponding point feature connected via a leader line. The combination of internal and external labeling techniques [11] and several enhanced variants [12], [13] have also been explored. Moreover, due to the recent development of digital maps available on mobile displays, scale-dependent consistent labeling techniques have also been investigated both in internal [14], [15] and external [16] labeling techniques. A heuristic algorithm has been developed by Haunert et al. [17], which requires users to carefully choose the scale of labels to find their proper placement. Furthermore, designers usually prefer to align leader lines along the octilinear directions [4], [18]. In our work, we present an internal labeling technique for placing station names in the octilinear layout of a railway network by extending the idea presented in [4].

III. Railway Map Schematization

We based our approach for progressive annotation on the previous formulation developed by Nöllenburg and Wolff [2], and we will briefly explain this formulation here.

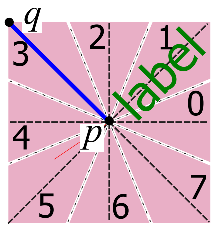
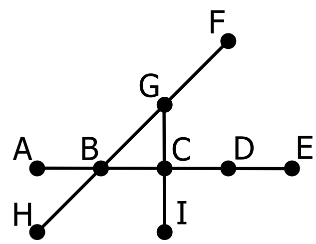


Fig. 2. The sector index of each octilinear direction ranges from 0 to 7. Here, the sector indices of $\mathrm{dir}(p, q)$ and "label" are 3 and 1, respectively.



	A	B	C	D	E	F	G	H	I
A	0	1	2	3	4	3	2	2	3
B	1	0	1	2	3	2	1	1	2
C	2	1	0	1	2	2	1	1	1
D	3	2	1	0	1	3	2	3	2
E	4	3	2	1	0	4	3	4	3
F	3	2	2	3	4	0	1	3	3
G	2	1	1	2	3	1	0	2	2
H	2	1	2	3	4	3	2	0	3
G	3	2	1	2	3	3	2	3	0
	20	13	12	17	24	21	14	20	19

Fig. 3. Computation of the sum of pairwise geodesic distances over the railway network. The element of the matrix on the right shows the respective pairwise shortest distance between the corresponding end vertices over the railway network on the left.

In practice, we take the geographic layout as input of the railway network and transform it into an octilinear layout using [2]'s formulation. We represent stations as vertices and their connections as edges in our implementation. The previous formulation [2] facilitates the solution of constrained optimization problems based on mixed-integer programming (MIP). This MIP-based formulation includes three different types of costs to be minimized, i.e., *line bends*, *relative positions*, and *edge length*.

The first cost, *line bends*, penalizes the number of bends along the respective railway lines. Let us define the line bends between the two edges $\overline{pq}$ and $\overline{qr}$ by referring to the angle spanned by the two edges $\angle pqr$. We set the corresponding cost of the line bends $\mathrm{bd}(p, q, r)$ to be 0 if $\overline{pqr}$ composes a completely straight line and increase it by an increment of 1 whenever the corresponding angle sharpens by 45 degrees. This implies that $\mathrm{bd}(p, q, r)$ ranges from 0 to 3. The total cost of the line bends can be written as:

$$c_{(\mathrm{B})} = \sum_{pqr \in L} \mathrm{bd}(p, q, r) \tag{1}$$

where L represents a set of three consecutive vertices along the respective railway lines.

The second cost, *relative positions*, evaluates the dissimilarity in the directions of the railway network edges between the geographic and schematic layouts. This is introduced to faithfully simulate the geographic shape of the input railway network in the schematized version. Suppose that the railway network edge $\overline{pq}$ originally falls into one of the eight sectors illustrated in Figure 2, and it is aligned to one of the octilinear directions through the schematization process. We denote by the difference between the original sector and the aligned octilinear direction $\mathrm{rpos}(p, q)$. The total cost of the relative positions is

$$c_{(\mathrm{P})} = \sum_{pq \in E} \mathrm{rpos}(p, q), \tag{2}$$

where E represents the set of edges in the railway network.

The last cost, which is referred to as *edge length*, is prepared for minimizing the total length of the railway network edges. This means that the cost allows us to make the entire map as compact as possible by maximally contracting total edge length as short as possible. This is accomplished by minimizing the following cost:

$$c_{(\mathrm{L})} = \sum_{pq \in E} \lambda(p, q), \tag{3}$$

where $\lambda(p, q)$ represents the length of the edge $\overline{pq}$ and E is again the set of railway network edges on the map. In this way, we can penalize unnecessary stretch of the edges in this formulation. Note that we impose the lower bound for the variable $\lambda(p, q)$, where D is set to be 1.0 by default.

In this formulation, the total cost is defined as a weighted sum of these three types of costs as:

$$c_{(\mathrm{total})} = w_{(\mathrm{B})} c_{(\mathrm{B})} + w_{(\mathrm{P})} c_{(\mathrm{P})} + w_{(\mathrm{L})} c_{(\mathrm{L})}, \tag{4}$$

where $w_{(\mathrm{B})}$, $w_{(\mathrm{P})}$, and $w_{(\mathrm{L})}$ are the weights for the three costs, respectively. In our implementation, we set $w_{(\mathrm{B})} : w_{(\mathrm{P})} : w_{(\mathrm{L})} = 1 : 1 : 100$, by default.

Refer to the paper [2] for more details about the constraints for aligning railway network edges to the octilinear directions while preserving the relative geographic positions and avoiding unwanted conflicts among railway lines. Using this formulation, we can obtain the initial schematic layouts of railway networks as shown on the top-left of Figure 1, Figure 4(a), and in the top rows of Figures 5 and 6.

IV. Progressive Annotation of Railway Stations

Once we have schematized the input railway network, the next step is to place a station name label in the neighborhood of each station vertex. As described earlier, the central downtown area of a railway network is inherently crowded with multiple railway lines and interchange stations, and thus it is often the case that we cannot find enough labeling space around the stations in that area. The problem becomes even harder if we want to align the name labels along the underlying octilinear directions around the station vertices. This process is difficult primarily because we have to explore a large number of label alignment patterns over the entire railway map under a complicated set of constraints.

To solve this problem, we have invented a novel algorithm for annotating station vertices progressively from the map center to its borders. Here, we assume that we take as input the octilinear layout of the railway map, which is obtained using the aforementioned MIP-based formulation [2].

A. Sorting station vertices for progressive annotation

It is plausible to explore the best placement of station name labels simultaneously over the entire railway map. However, this usually results in a large search space for the optimal solution and takes a considerable amount of computation time. Thus, we have to reduce this search space to make the annotation problem feasible in terms of the computation time. Our idea is to locate the optimal placement of the annotation labels by moving progressively from the crowded central downtown area to sparse rural areas. This is achieved by sorting the station vertices according to a specific distance measure defined within the railway network and finding the optimal alignment of their labels one by one.

To obtain the distance measure, we first sum up pairwise geodesic (i.e., shortest) distances for the respective station vertices over the railway network. This measure is inspired by previous work [19], which was originally developed for formulating the shape descriptor of 3D meshes. Figure 3 shows such an example, where each element of the right matrix represents the shortest distance between the corresponding vertices over the network on the left, which can be obtained using the Dijkstra algorithm. The bottom row of the distance matrix lists the sum of the pairwise distances for each vertex. We then sort the station vertices in ascending order according to the distance sum, which means that we arrange the interchange vertices C, B, and G as the first three elements in the sorted sequence. In our formulation, this measure successfully allows us to identify interchange stations in the central downtown area as vertices that appear earlier in the sorted sequence. This enables us to effectively place annotation labels in the dense downtown area first and then move on to those in sparse rural areas, while keeping the total computation cost sufficiently reasonable.

B. Octilinear alignment of annotation labels

We formulated our new annotation algorithm based on two design rules. The first rule is that we increase the labeling space when necessary by elongating the railway network edges while retaining their original octilinear directions. The second rule is that we place each label sufficiently close to the corresponding vertex in such a way that it is aligned with one of the eight octilinear directions. For selecting the best placement of each label, we explore possible directions of the label so that it makes the largest angle with existing railway lines (e.g., $\overline{pq}$) emanating from the station vertex p, as shown in Figure 2. According to [4], this can be formulated by optimizing $\beta(p,q)$ that satisfy the following equation:

$$\min\{|\mathrm{dir}(p,q)-\alpha, 8-|\mathrm{dir}(p,q)-\alpha|\} = 4-\beta(p,q)$$

where $\mathrm{dir}(p,q)$ and α represent sector indices of $\overline{pq}$ and the label, respectively. This implies that we need to introduce a new cost term

$$c_{(\mathrm{D})} = \sum_{p\in V, \overline{pq}\in E_p} \beta(p,q)$$

into our formulation, where V and E represent the sets of station vertices and their incident edges, respectively. The total cost Eq. (4) can be replaced with

$$c_{(\mathrm{total})} = w_{(\mathrm{B})}c_{(\mathrm{B})} + w_{(\mathrm{P})}c_{(\mathrm{P})} + w_{(\mathrm{L})}c_{(\mathrm{L})} + w_{(\mathrm{D})}c_{(\mathrm{D})}, \quad (5)$$

where $w_{(\mathrm{B})} : w_{(\mathrm{P})} : w_{(\mathrm{L})} = 1 : 1 : 100 : 10$ by default.

TABLE I
COMPUTATION TIMES (IN SECONDS).

map	time	map	time	map	time
Figure 1	22.3	Figure 5(a)	22.5	Figure 6(a)	76.3
Figure 4(b)	42.2	Figure 5(b)	23.1	Figure 6(b)	122.6
Figure 4(c)	33.3				

In our algorithm, we compute the optimal alignment of station name labels one by one by referring to the aforementioned order of vertices based on the sum of the pairwise topological distances. Here, we adjust the size of each label by counting the number of characters in the corresponding station name. When we find unwanted overlaps among railway network edges and labels, we additionally incorporate constraints that eliminate such conflicts and recompute the optimization problem until we can find a conflict-free layout. This effectively allows us to properly elongate the railway network edges to spare minimum necessary space for placing the label. Note that once we stretch an edge in the railway network, we replace its lower bound with the new length and employ this new length when optimizing the cost in Eq. (3). This helps us adaptively expand the size of the entire railway network every time we insert a new label. This annotation process is carried out individually for each station vertex until all name labels are properly placed. Figure 1 shows this process for progressively annotating stations over a schematic railway map.

V. RESULTS

Our prototype system has been implemented on a desktop PC with 2.7 GHz 12-Core Intel Xeon E5 CPU and 64GB RAM. The source code was written in C++ using OpenGL for drawing maps, Boost Graph Library for constructing the graph data structure, CPLEX for the MIP-based optimization, and GLUI for the user interface. Table I shows computation times necessary for schematizing existing railway maps, which was obtained by averaging three trials.

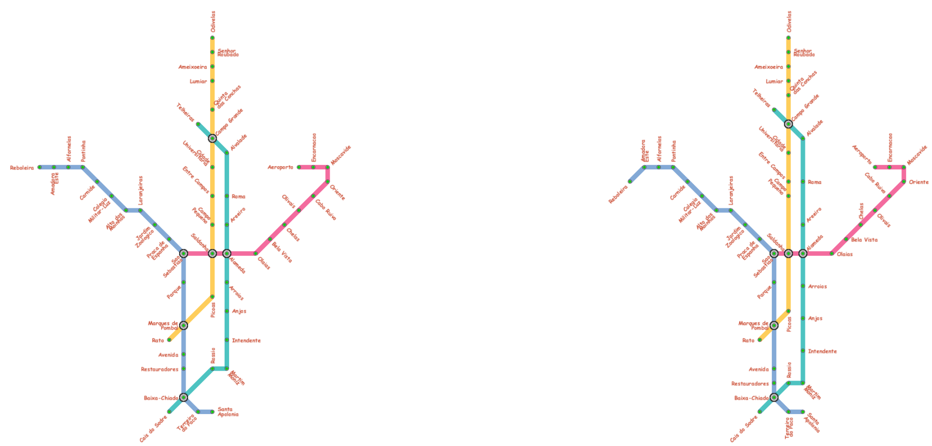
Figure 1 presents a sequence of the metro maps of Ryon, in which we progressively introduce station name labels one by one, from the central downtown area to the border rural areas. The results show that we can successfully spare additional space for each incoming label by elongating the railway network edges while retaining their directions.

Figure 4 exhibits the metro maps of Montreal, where we set $w_{(\mathrm{B})} : w_{(\mathrm{P})} : w_{(\mathrm{L})} : w_{(\mathrm{D})} = 1 : 1 : 100 : 1$ in Figure 4(a) while $w_{(\mathrm{B})} : w_{(\mathrm{P})} : w_{(\mathrm{L})} : w_{(\mathrm{D})} = 1 : 1 : 100 : 100$ in Figure 4(b). The figure shows that, as we increase the weight $w_{(\mathrm{D})}$, we can align the station name labels to be more perpendicular to the railway lines. Raising this weight value also forces us to spare more labeling space around the downtown area in an early stage of the optimization and consequently results in less computation time as a whole.

We can also change the constraints over the coordinates of station vertices and edge lengths in the metro maps of Lisbon. In Figure 5(a), we restrict the station vertices to be at the integer grid coordinates. Conversely, in Figure 5(b), we relaxed the coordinates of station vertices and edge lengths to real

(a) (b) (c)

Fig. 4. Montreal metro map: (a) Schematic metro map. (b) Annotated metro map with a low weight for the label alignment ($w_{(\mathrm{B})} : w_{(\mathrm{P})} : w_{(\mathrm{L})} : w_{(\mathrm{D})} = 1 : 1 : 100 : 1$). (c) Annotated metro map with a high weight for label alignment ($w_{(\mathrm{B})} : w_{(\mathrm{P})} : w_{(\mathrm{L})} : w_{(\mathrm{D})} = 1 : 1 : 100 : 100$).



(a) (b)

Fig. 5. Side-by-side comparison between Lisbon metro maps. (a) Station vertices are restricted to be on the grid integer coordinates. (b) Station vertices are allowed to have real-valued coordinates.

values. The side-by-side comparison reveals that we can keep the entire map compact without introducing redundant labeling space when we employ real-valued coordinates, although we may locally lose balanced placement of annotation labels. In our implementation, we employ real-valued coordinates for more compact layouts of annotated railway maps.

Figure 6 presents more complex examples, where Figures 6(a) and (b) correspond to annotated metro maps in Vienna and Taipei, respectively. These results demonstrate that our approach can successfully spare enough labeling space in the crowded downtown area by elongating the railway network edges while keeping their original directions.

VI. Conclusion

In this paper, we have presented an approach to progressively annotating stations in schematic railway maps. The approach allows us to automatically place name labels in the neighborhoods of stations one by one from the crowded downtown area to the border rural areas. This is accomplished by elongating the railway network edges properly to spare the necessary labeling space while retaining their original directions in the schematic railway map. Furthermore, we formulated our approach in a way that aligns the station name labels with one of the octilinear directions by taking advantage of the conventional map schematization technique. Experimental results present the feasibility of the proposed approach with several design examples of annotated metro maps in major cities.

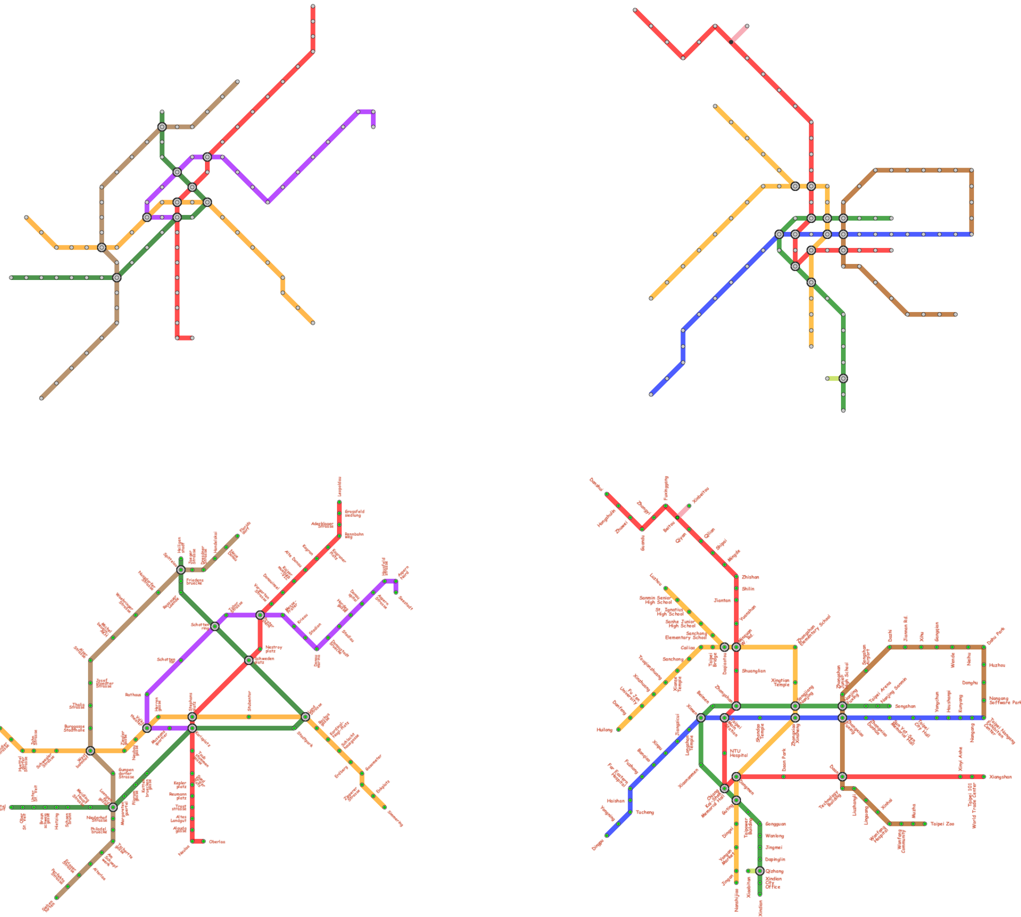
Our future work includes placing station name labels consistently along each railway line in order to explore more aesthetic and compact map layouts. For this purpose, clustering stations based on the map context is important. Another interesting theme for future research is to explore a well-balanced map layout while stretching its spatial domain in the progressive annotation process.

Acknowledgments

This work as been partially supported by JSPS KAKENHI (grant numbers 16H02825, 26730061, and 15K12032). This work has been also conducted under the support by Joint Research Program No. 398 at CSIS, the University of Tokyo.

References

[1] J. Stott, P. Rodgers, J. C. Martinez-Ovando, and S. G. Walker, "Automatic metro map layout using multicriteria optimization," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 1, pp. 101–114, 2011.



(a) (b)

Fig. 6. Annotating complex railway networks. (a) Vienna metro map. (b) Taipei metro map. The top and bottom rows correspond to input schematic layouts and annotated layouts obtained using our approach, respectively.

[2] M. Nöllenburg and A. Wolff, "Drawing and labeling high-quality metro maps by mixed-integer programming," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 5, pp. 626–641, 2011.

[3] Y.-S. Wang and M.-T. Chi, "Focus+context metro maps," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2528–2535, 2011.

[4] H.-Y. Wu, S. Takahashi, D. Hirono, M. Arikawa, C.-C. Lin, and H.-C. Yen, "Spatially efficient design of annotated metro maps," *Computer Graphics Forum*, vol. 32, no. 3, pp. 261–270, 2013.

[5] H.-Y. Wu, S. Takahashi, C.-C. Lin, and H.-C. Yen, "Travel-route-centered metro map layout and annotation," *Computer Graphics Forum*, vol. 31, no. 3, pp. 925–934, 2012.

[6] H. Y. Wu, S. H. Poon, S. Takahashi, M. Arikawa, C. C. Lin, and H. C. Yen, "Designing and annotating metro maps with loop lines," in *19th International Conf. Information Visualisation*, 2015, pp. 9–14.

[7] P. Claudio and S.-E. Yoon, "Metro transit-centric visualization for city tour planning," *Computer Graphics Forum*, vol. 33, no. 3, pp. 271–280, 2014.

[8] J. Christensen, J. Marks, and S. Shieber, "An empirical study of algorithms for point-feature label placement," *ACM Trans. Graphics*, vol. 14, no. 3, pp. 203–232, 1995.

[9] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff, "Boundary labeling: models and efficient algorithms for rectangular maps," *Computational Geometry: Theory and Applications*, vol. 36, no. 3, pp. 215–236, 2007.

[10] C.-C. Lin, "Crossing-free many-to-one boundary labeling with hyper-leaders," in *Proceeding of the 3rd IEEE Pacific Visualization Symp. (PacificVis 2010)*, 2010, pp. 185–192.

[11] H.-Y. Wu, S. Takahashi, C.-C. Lin, and H.-C. Yen, "A zone-based approach for placing annotation labels on metro maps," in *Proc. of the 11th International Conf. Smart Graphics*, ser. Springer Lecture Notes in Computer Science, vol. 6815, 2011, pp. 91–102.

[12] J.-D. Fekete and C. Plaisant, "Excentric labeling: Dynamic neighborhood labeling for data visualization," in *Proc. of the SIGCHI conference on Human factors in computing systems*, 1999, pp. 512–519.

[13] M. Fink, J.-H. Haunert, A. Schulz, J. Spoerhase, and A. Wolff, "Algorithms for labeling focus regions," *IEEE Trans. Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2583–2592, 2012.

[14] K. Been, E. Daiches, and C. Yap, "Dynamic map labeling," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 5, pp. 773–780, 2006.

[15] H.-Y. Wu, S. Takahashi, S.-H. Poon, and M. Arikawa, "Scale-Adaptive Placement of Hierarchical Map Labels," in *EuroVis 2017 - Short Papers*. The Eurographics Association, 2017.

[16] ——, "Introducing leader lines into scale-aware consistent labeling," in *Advances in Cartography and GIScience*, M. P. Peterson, Ed. Springer, 2017, pp. 117–130.

[17] J.-H. Haunert and B. Niedermann, "An algorithmic framework for labeling network maps," in *Computing and Combinatorics*, D. Xu, D. Du, and D. Du, Eds., 2015, pp. 689–700.

[18] M. A. Bekos, M. Kaufmann, M. Nöllenburg, and A. Symvonis, "Boundary labeling with octilinear leaders," *Algorithmica*, vol. 57, no. 3, pp. 436–461, 2010.

[19] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D models with shape distributions," in *Proc. of International Conf. Shape Modeling and Applications*, 2001, pp. 154–166.