DOI: 10.1111/j.1467-8659.2012.03085.x
Eurographics Conference on Visualization (EuroVis) 2012
S. Bruckner, S. Miksch, and H. Pfister
(Guest Editors)

Volume 31 (2012), Number 3

Travel-Route-Centered Metro Map Layout and Annotation

Hsiang-Yun Wu[1], Shigeo Takahashi[1], Chun-Cheng Lin[2], and Hsu-Chun Yen[3]

[1]The University of Tokyo, Japan
[2]National Chiao Tung University, Taiwan
[3]National Taiwan University, Taiwan


(a) (b) (c)

Figure 1: *Vienna metro map. (a) Geographical layout. (b) Conventional MIP layout. (c) Customized layout using our approach.*

Abstract

When providing travel guides for a specific route in a metro network, we often place the route around the center of the map and annotate stations on the route with thumbnail photographs. Nonetheless, existing methods do not offer an effective means of customizing the network layout in order to accommodate such large annotation labels while preserving its planar embedding. This paper presents a new approach for designing the metro map layout in order to annotate stations on a specific travel route with large annotation labels. Our idea is to elongate the travel route to be straight along the centerline of the map so that we can systematically annotate such stations with external labels. This is accomplished by extending the conventional mixed-integer programming technique for computing octilinear layouts where orientations inherent to the metro line segments are plausibly rearranged. The stations are then connected with external labels through leaders while minimizing intersections with metro lines for enhancing visual clarity. We present several design examples of metro maps and user studies to demonstrate that the proposed aesthetic criteria successfully direct viewers' attention to specific travel routes.

1. Introduction

As is often the case with complex metro maps, finding a specific route on a metro map is a tiresome task, especially when we have to travel in an unfamiliar city. Customized map layouts usually help us to find such travel routes since they direct our attention to these routes even within the global context view of the map. In this case, we commonly deform the layout of the metro network so that we can make the route straight and place it along the centerline of the map domain, as in straight-line diagrams (see Figure 2 for example), which we usually find inside of metro carriages or on pamphlets prepared by the metro companies. The same strategy is also employed in travel guide maps available at bookstores. Passengers found this type of metro map layout useful not only for effectively finding a specific path embedded in the layout [WPCM02], but also for directing viewers' first attention toward that path. Furthermore, we commonly introduce large annotation labels containing texts and photographs for providing more information about the stations on the route, in order to facilitate travelers to ex-

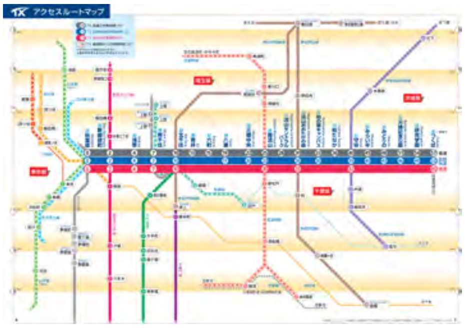


Figure 2: *A route-aware map of Tsukuba Express (courtesy of Metropolitan Intercity Railway Company, Japan).*

plore their own travel routes. In practice, this type of map layout has originated from cartographic maps called *strip maps*, where a highway route together with its surrounding landmarks is elongated and drawn horizontally or vertically [Mac86, MJ87]. This lasting use of such line-driven map representations [AS01] reveals the importance of this type of route-centered map layout.

On the other hand, since Beck devised schematic representations of metro networks in 1933 [Rob03], his design principle has been most commonly used for drawing metro networks where metro lines are aligned with octilinear directions such as horizontal, vertical, and diagonal while preserving their geographical embeddings. This principle provides perceptually plausible criteria in the sense that our visual system is the most sensitive to patterns along the octilinear directions. Nonetheless, it is still technically challenging to develop an algorithm for composing octilinear layouts of route-centered metro maps since we have to incorporate necessary deformation while optimizing their layout under the octilinearity constraints. The problem becomes further complicated when trying to incorporate large annotation labels since existing techniques tackle map layout and external label placement problems individually.

This paper presents an approach for designing metro map layouts in order to annotate stations on a specific travel route with large annotation labels. Our idea is to transform a geographical metro map into its octilinear layout while elongating the travel route to be straight along the centerline of the map domain. This is achieved by adaptively adjusting the orientations of metro line segments in the conventional mixed-integer programming formulation. The travel-route-centered layout allows us to systematically annotate the stations on the route with large *external* labels such as thumbnail photographs and explanation texts. The *minimum cost maximum flow problem* formulation has been employed to connect stations and labeling boxes with leaders while minimizing their intersections with the metro network. Our technical contribution lies in the new formulation of computational algorithms for customizing the layouts of travel-route-centered metro maps concurrently with placement of large annotation labels.

Our scenario for designing metro maps is summarized as follows: Our metro map design begins with specifying a travel route in the metro network so that our system can change the orientation of the entire metro map, and then elongate the route to be horizontally straight. We also equip our design system with an interface for interactively adjusting the orientation of the metro network and its local shapes, which allows us to refine the travel-route-centered layout by trial and error. The stations on the route are then annotated automatically with large external labels using *boundary labeling*. Here, the shapes of the label leaders are systematically rearranged for avoiding excessive intersections with metro lines within the map content.

The remainder of this paper is organized as follows: Section 2 provides a survey on relevant techniques for the design of metro map layouts and their annotation. Section 3 describes our method for designing a travel-route-centered metro map while retaining its octilinear layout. Section 4 details an algorithm for annotating stations on the travel route while minimizing intersections between the label leaders and metro lines. Section 5 presents several design examples together with user studies to demonstrate the feasibility of our approach, which is followed by conclusions and future work in Section 6.

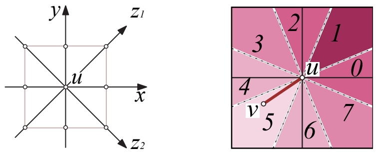
2. Related Work

A graph is a ubiquitous and fundamental data structure for visualizing a schematic metro network since we can represent metro stations as its nodes and metro lines as its edges. Extensive studies have been conducted for preserving specific graph layouts and their planar embeddings as well as minimizing intersections between edges. As for metro networks, aesthetic criteria invented by Beck are now considered as a standard practice where metro lines are aligned to diagonal directions at 45 degrees as well as horizontal and vertical ones [Rob03]. This graph layout provides a visually pleasing representation of a metro network since our eyes are sensitive to the patterns along these directions. Nonetheless, this usually results in constrained optimization problems and thus limits the degree of freedom in arranging the metro network on the map domain [Wol07].

Pioneering work was done by Hong et al. [HMD05], where they developed an algorithm for automatically drawing metro networks in an aesthetic layout. In their algorithm, the spring-embedder model has been altered by incorporating additional magnetic forces so that the metro lines can be aligned to octilinear directions. Stott et al. [SRMOW11] introduced a method for optimizing the fitness of a metro network layout using a hill climbing optimization technique while preserving a predefined set of aesthetic criteria. Their criteria include the weighted sum of angular resolution at nodes, edge length uniformity, edge length balancing, line straightness, and edge octilinearity, while unfortunately the octilinearity has not been rigorously maintained in the case of complex network structures. On the other hand, Nöllenburg and Wolff presented an approach that faithfully respects the octilinear layout of a metro network by introducing mixed-integer programming (MIP) [NW11]. They formulated the aesthetic criteria as linear equality and inequality constraints and optimized the linear fitness function representing the number of line bends, total edge length, and changes in the edge orientations. Recently, accelerated computation for octilinear metro map layouts has been accomplished by Wang and Chi [WC11], which allows us to interactively enlarge a specific route through map deformation. Their optimization scheme puts the first priority on visual clarity of the selected route while suppressing the rest of the metro network by fading its color. Several related techniques such as path simplification [MG06,DHM08] and line-crossing minimization [BKPS08, ABKS10] have also been proposed.

Annotation labels are used to assign additional information to map landmarks called *sites*, and play another important role in the aesthetic design of metro maps. Famous annotation problems such as the label-size maximization problem and label-number maximization problem are known to be NP-hard [WWKS01, KI03]. Technically, annotation labels are categorized into two groups: *internal labels* that are placed close enough to the sites and *external labels* that are often used for placing large annotation labels and thus are placed sufficiently far away from the sites. For placing internal labels, for example, a force-directed model can be employed to minimize the distance between the site and its corresponding annotation label [SSB06]. On the other hand, different metrics should be incorporated for arranging external labels since their placement has a significant influence on the visual quality of the map layout. Among such external labeling techniques, Bekos et al. introduced the theoretical formulation for the concept of boundary labeling [BKSW07], where the external labels are placed in the boundary margin around the map content area. Another concern for boundary label placement is how to draw leaders between the sites and labeling boxes in a visually pleasing manner. Drawing leaders along the octilinear directions [BKNS10] again helps us to augment the visual clarity of the map annotation while we may not be able to avoid intersections between leaders and map content due to limited flexibility in leader shapes. Different leader shapes have been introduced for this purpose together with several theoretical bounds on computational complexity [LKY08, Lin10, BKNS10]. A hybrid approach has also been invented for placing both internal and external labels at the same time [BKPS11, WTLY11].

Currently available schematic designs in cartography motivate us to customize geographical maps according to our own travel purposes. One of the typical examples is a *strip map*, which has been originally used for highlighting a specific route in the travel throughout recorded history, and currently serves as a guide for us to locate our current positions in travels [MJ87, Mac86]. This idea has been incorporated into the concept of route-aware maps [AS01, SRP10], which effectively guide travelers with schematic representations of the travel routes and their surrounding landmarks only. The concept of travel-route-centered metro maps has been inspired by these cartographic map designs, while it is more computationally involved since we seek the compromise between visual clarity of the route in the global map context and aesthetic placement of annotation labels in accordance with the octilinear layout of a metro network.



(a) (b)

Figure 3: *Aligning metro lines with octilinear directions. (a) Octilinear directions emanating from the node u, and (b) the corresponding fan-shaped sectors with respect to u.*

3. MIP-Based Design of Metro Layouts

This section describes how we transform a specific travel route to be straight along the centerline of the map while preserving the octilinear layout of the entire metro network.

3.1. MIP formulation for obtaining octilinear layout

For obtaining octilinear layouts of metro networks, we employ the aforementioned MIP formulation [NW11] in our approach, because this formulation provides us with sufficient degrees of freedom to control the shape of the map layout with additional constraints as opposed to the conventional methods [HMD05, SRMOW11, WC11]. Indeed, the MIP formulation encodes several design rules as linear equality and inequality constraints in order to rigorously align metro lines with octilinear directions, while retaining the circular order of incident edges at each node.

While enforcing these constraints on the metro layout, the approach optimizes the linear fitness function for minimizing the number of line bends, total edge length, and difference in edge orientation from the original geographical layout. Note that, among these three fitness values, the last one has a significant influence on the resulting metro layout while the first two are basically independent. Roughly speaking, our idea is to edit such orientations inherent to metro edge segments so that we can generate a specific travel-route-centered layout within the MIP framework.

In the conventional MIP formulation [NW11], all edge segments are guaranteed to be aligned to one of the eight

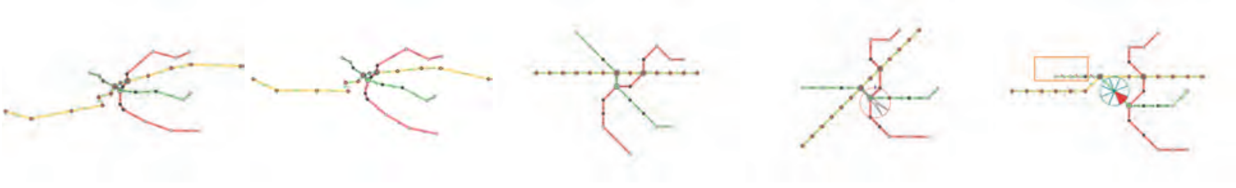
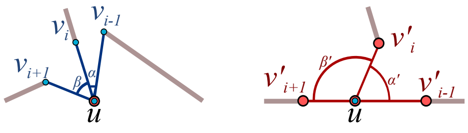

(a) (b) (c) (d) (e)

Figure 4: *An example scenario for designing a travel-route-centered map layout. (a) Selecting a specific route. (b) Rotating the entire map by referring to the route. (c) Route-centered octilinear layout. (d) Rotating the entire metro map again. (e) Rearranging a local set of edge orientations.*



(a) (b)

Figure 5: *Adjusting edge orientations. (a) Original layout of four stations and (b) its updated layout after the changes in edge orientation are propagated.*

octilinear directions as shown in Figure 3(a). This is done first by assigning an integer number to each of the eight fan-shaped sectors as shown in Figure 3(b) and then classifying each edge segment into an appropriate sector according to its orientation in the original geographical map. For example, the orientation of the edge $\overline{uv}$ shown in Figure 3(b) can be encoded as 5. The conventional MIP formulation allows us to select one final edge orientation from three octilinear directions, i.e. the original directions and its two neighboring directions. This is because the MIP formulation maximizes the similarity of each edge orientation to its original one so as to effectively simulate the geographical layout of a metro network with an octilinear version. Again, in this case, the set of possible orientations for the edge $\overline{uv}$ are 4, 5, and 6 in the conventional MIP formulation. In our approach, on the other hand, we prepare an additional angle variable for each edge and refer to the variables when generating the octilinear layout with the conventional MIP formulation. This gives us enough degree of freedom to explore the route-centered layout of the metro network by adjusting such angle variables of the metro edges on the route. Readers can refer to the details of the MIP mathematical formulation in [NW11].

3.2. Elongating travel routes along the centerline

We are now ready to describe how to make the specific travel route straight in the map through the MIP optimization process. First we expect a user to specify a travel route, for example, by clicking on the departure and destination stations in the geographical map. The route is automatically identified through Dijkstra's shortest path algorithm as shown in Figure 4(a). By taking such a specific travel route as input, we update the angle variables of all the edges on the route so that the route extends out horizontally straight in the map. Our design scenario achieves this goal first by rotating the overall metro network in order to lay out the two endpoints of the route horizontally in the map, as shown in Figure 4(b). This step is then followed by changing the angle variables of the route edges to align them along the horizontal centerline of the map domain. With this configuration, the MIP optimizer can elongate the travel route to be horizontally straight while retaining the overall octilinear layout and original planar embedding of the metro network, as shown in Figure 4(c).

Nonetheless, simply adjusting edge orientations on the route sometimes fails to make the route straight or incurs unsolvable cases. This is because orientations of the edges next to the route still remain the same, and thus usually incur large gaps from the orientations of the edges along the route. In our approach, this is alleviated by propagating the changes in edge orientation on the route to the remainder of the metro network, while respecting the original relative positioning of metro stations and lines. This process proceeds as follows. Suppose that four station nodes u, v_{i-1}, v_i, and v_{i+1} are originally laid out as shown in Figure 5(a), where v_{i-1}, v_i, and v_{i+1} are adjacent to the node u and ordered counter-clockwise around u. The orientations of edges $\overline{uv_{i-1}}$ and $\overline{uv_{i+1}}$ are updated as $\overline{uv'_{i-1}}$ and $\overline{uv'_{i+1}}$ through adjusting the angle variables of the edges on the route. We change the orientation of edge $\overline{uv_i}$ by a small amount so that it becomes closer to $\overline{uv'_i}$ while equalizing the relative angle ratio $\alpha' : \beta'$ with the original ratio $\alpha : \beta$. We apply this operation iteratively to all edges until convergence while we fix the edge orientations on the route during the propagation process. This significantly facilitates us to solve the corresponding MIP problem also, since the edge orientations smoothly change over the metro network after this propagation process. Furthermore, this treatment of edge orientations usually allows us to find plausible solutions at early stages of the MIP optimization process since the modulated edge orientations are more likely to relax constraints for finding an octilinear metro layout.

3.3. Interactive editing of metro layouts

As a post-process, a user can manually rearrange the octilinear layout of the metro map in our implementation. In practice, we equipped our prototype system with an interface so that we can conduct a two-step design scenario. Suppose that we are currently modifying the route-centered octilinear layout of the metro map as shown in Figure 4(c). In our scenario, we first change the overall orientation of the metro network with respect to the center of the map domain. As shown in Figure 4(d), a dial metaphor appears at the center when rotating the metro network so that we can visually confirm the rotation angle that we are going to apply. We then move on to the orientation adjustment of a local set of edges in order to rearrange the relative positioning between metro stations and lines. For this task, we select a set of edges locally by using a rubber-band metaphor as shown in Figure 4(e), and change their rotation angles using the eight slices pie metaphor. This design strategy is effective especially when we edit star-shaped metro networks that typically exist in major cities, because we are more likely to keep the shape of the metro network around the central downtown area while applying drastic changes to those in the suburb areas. Users can apply MIP optimization any time when they want to see the resulting octilinear layout of the metro map with the current configuration of the edge orientations (Figure 4(e)). Note that the angle variables of route edges are fixed so that the associated edge orientations remain to be horizontal. The overall design can be carried out by trial and error until the user can obtain satisfactory results.

4. External Label Placement Design

Having obtained a route-centered layout of the metro map, we place external labels to annotate stations on the route with large labels such as thumbnail photographs. For this purpose, we developed a new flow-network-based algorithm for placing such external labels while minimizing the intersections between label leaders and metro lines.

4.1. Requirements for arranging labels and leaders

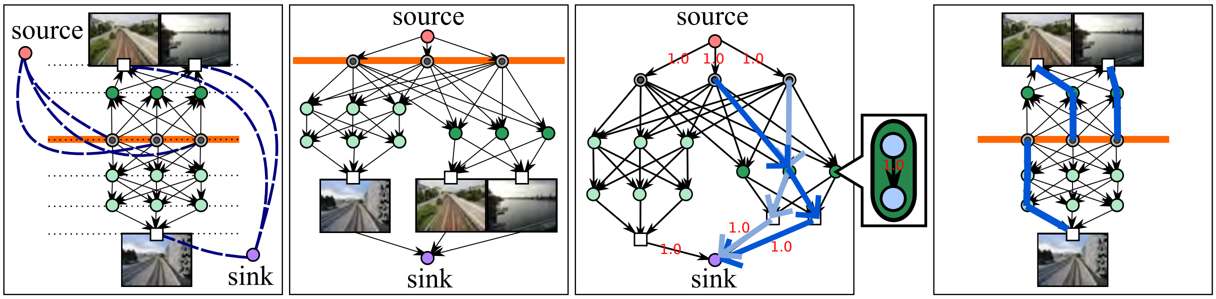
In our approach, we employ the *boundary labeling* technique [BKSW07] for aligning external labels horizontally in the marginal space around the map content area. This is because we can systematically determine the positions of the external labels while avoiding excessively wasting space in the map domain. One of the most challenging problems in conventional boundary labeling techniques has been the appropriate design of leader shapes while minimizing the number of intersections between every pair of leaders, as well as the total leader length. To make matters worse, in our case, we also have to consider possible intersections of leaders with map contents such as metro lines, to make sure that the metro network visually pops up on top of the map content. Indeed, incorporating this aesthetic criteria significantly improves the readability of the metro map itself, and effectively reduces visual clutter arising from line intersections (cf. Section 5). As described in Section 2, conventional techniques for boundary labeling employ a straight line segment or line segment aligned with orthogonal/octilinear directions as a label leader. In our case, the metro network has been already arranged to octilinear directions, and thus it may visually conflict with label leaders if they are also aligned with octilinear directions. This leads us to the idea of introducing curved segments as label leaders to our approach. Actually, this scheme helps users to focus their attention on the main route of the metro network while suppressing the influence of the label leaders on its map readability.

4.2. Constructing flow networks for designing leaders

As described earlier, conventional boundary labeling techniques determine the layout of external labels only by taking into account mutual intersections between label leaders, and never explicitly avoid intersections with important landmarks in the map content. However, it is true again that a metro map consists of line segments, and the associated connectivity of metro lines has significant information for travelers. Thus, we should minimize the number of intersections between leaders and metro lines also to enhance the visual readability of the metro map itself. We formulate this problem as a combination of *bipartite matching* problems, then reduce this composite problem to a general network flow problem, and finally find its *minimum cost maximum flow* by taking advantage of the well-known *successive shortest path algorithm* in [EK72].

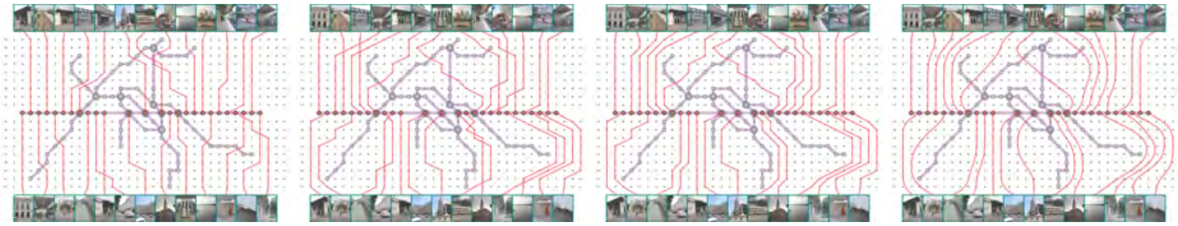
Figure 6(a) shows how we construct the flow network over the map domain. Our leader computation starts with locating *intermediate nodes* of the flow network F, where nodes are placed on grid samples that coincide with integer coordinates for computing the octilinear metro layout. In Figure 6(a), two different groups of intermediate nodes, which are colored in dark and light green, are placed above and below the specified route except for the positions of metro stations that are not contained in the selected route. Note that the orange line represents the selected route on the metro network, the gray circles correspond to the station nodes on the route, and white squares represent *port nodes* of annotation labels.

The connectivity of the flow network can be established by connecting each intermediate node with every node in its next horizontal rows, while the station nodes and label port nodes are connected with all the intermediate nodes along their immediate next rows. This is equivalent to composing a multi-level bipartite network between intermediate nodes of two adjacent horizontal rows, where each pair of the two adjacent rows constitutes a complete bipartite graph inside the entire flow network. We also introduce virtual source and sink nodes to the composite bipartite network and connect them with station nodes and label port nodes, respectively, by following the conventional strategy for transforming a



(a) (b) (c) (d)

Figure 6: *Designing leader shapes using flow networks. (a) Network construction in the map domain. (b) Topologically rearranged structure of the network. (c) Each intermediate node consists of two virtual nodes and an implicit edge with weight 1.0 in between. (d) Final flow paths for laying out leaders between stations and external labels.*



(a) (b) (c) (d)

Figure 7: *Effect of edge weights on leader shapes: (a) Only with square root of the difference along the horizontal axis as the edge weight. (b) Penalty weights are assigned to unwanted intersections between leaders and metro lines. (c) A pair of leaders having mutual intersection is swapped. (d) Endpoint-interpolating B-splines are employed for smoothing the leader shapes.*

simple bipartite matching to a network flow problem. For ease of explanation, we rearrange the topological structure of the flow network to clarify that the flow goes through the network from the top source node to the bottom sink node as shown in Figure 6(b), and orient each edge of the network to follow the flow direction as shown in Figure 6(c).

4.3. Minimum cost maximum flow computation

For finding the set of optimal flow paths that guide the shapes of label leaders, we have to solve the minimum cost maximum flow problem as described above. For this purpose, we prepare a residual network R where the weight of each edge represents an available flow capacity, in addition to the flow network F where the weight represents the actual amount of flow. Before exploring a set of optimal flows over the network, all the edge weights of F are initialized to be 0.0 while those of R will be adaptively adjusted to seek the aesthetic layout of label leaders, as will be described in Section 4.4.

We then use the successive shortest path algorithm [EK72], where we successively explore the optimal flow path over the network and update the flow and residual networks accordingly until we cannot find any valid path in the network. In our implementation, we first search for the maximum flow augmenting path from the source to the sink through the residual network R. We then negate the weights and inverse the orientations of the traversed edges in R so that the next optimal flow will not pass through these edges, and increase the flow weights of the corresponding edges in F by the amount of flow. This finally gives us a set of optimal flow paths that successfully guide aesthetic shapes of label leaders. For finding each maximum flow augmenting path, we employ the Bellman-Ford algorithm instead of Dijkstra's algorithm since the residual network R may retain negative weight values for its edges.

Note that in our weight assignment, we handle the edges incident to the *source* and *sink* nodes as a special case. Actually, we assign the weights 1.0 to these edges (Figure 6(c)), so that they serve as bottlenecks that bound the maximum amount of every possible flow path to be 1.0. Furthermore, the weights of other edges, as described in Section 4.4, are set to be at least more than 1.0, so the maximum amount of flow in the network is always equal to the number of annotated stations according to the max-flow min-cut theorem. In this case, every time when we find the shortest path, we can always increase the amount of flow by 1.0 due to the bottlenecks placed on the edges outgoing from the source and incoming to the sink. However, even with this setting, we cannot avoid cases where multiple flow paths intersect with each other at some intermediate nodes as shown in Figure 6(c). In order to guarantee that each intermediate node is visited at most once, we replace each intermediate node with two distinct nodes connected by an implicit edge, to which we again assign 1.0 as its weight as shown in Figure 6(c).

As for the computational complexity, the Bellman-Ford algorithm runs in $O(|V||E|)$ for finding a shortest path, where $|V|$ and $|E|$ indicate the numbers of nodes and edges in the network, respectively. In this way, we guide the shape of each label leader by a set of nodes contained in the corresponding flow path from the station node to the label port node. Figure 6(d) shows such an example, where the leaders are drawn as polylines by referring to the corresponding maximum flow augmenting path.

4.4. Avoiding intersections with metro lines

Our final task is to assign appropriate weights to edges connecting intermediate nodes in the residual network R to make label leaders satisfy two specific aesthetic criteria: the leader shapes should be as vertical as possible and their intersections with the underlying metro lines should be minimized. This is accomplished by defining the weight of a network edge e to be $w(e)$, which is given by

$$\begin{cases} p\sqrt{\dfrac{|\vec{e}\cdot\vec{x}|}{d}}+q\sum_{j}^{n}c_j(e)\cdot\left(\dfrac{\vec{e}\cdot\vec{l}_j}{|\vec{e}||\vec{l}_j|}+1\right)+1, & \text{if } |\vec{e}|<d \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

where d is the limit horizontal displacement ($d = 2$ on the integer coordinate system by default) and $l_j (j = 1,\ldots,n)$ is the j-th line segment of the octilinear metro network. $c_j(e)$ becomes 1.0 when $\vec{e}$ has intersection with $\vec{l}_j$ and 0.0 otherwise. Note that we penalize the length of $\vec{e}$ projected on the horizontal axis $\vec{x}$, so that the corresponding flow paths are more likely to be vertically straight. Here, the weight is set to be proportional to the square root of the horizontal displacement so that the penalty increases rapidly with the displacement. We also compute $c_j(e)$ by detecting the intersection between the edge $\vec{e}$ and metro line segment $\vec{l}_j$ in order to minimize the number of such intersections. In this formulation, we reduce the weight as the intersecting angle becomes more perpendicular by computing the inner product of $\vec{e}$ and $\vec{l}_j$, as shown in Eq. (1). The third term is added to guarantee that all edge weights are no less than 1.0, as described in Section 4.3.

Indeed, p and q are user-defined parameters that determine the relative importance of the two criteria in the weight function. Figure 7(a) shows a result where we draw the leaders as polylines just by connecting the sequence of nodes with line segments, which is obtained when $p = 10$ and $q = 0$ in Eq (1). In this case, we cannot avoid intersections between label leaders and metro lines, which may result in unwanted visual clutter. By increasing parameter q, we can penalize such intersections, and label leaders are more likely to bypass metro lines within the map domain. Figure 7(b) shows such an example where we set $p = 10$ and $q = 100$ to increase the weights of network edges that are intersecting with metro lines. On the other hand, this may introduce mutual intersections between label leaders as shown in the figure. We alleviate this problem by swapping label leaders having mutual intersections as shown in Figure 7(c). Finally, we smooth the shape of the leaders by employing endpoint-interpolating B-spline interpolation as shown in Figure 7(d), where the intermediate nodes on the polyline leader are used as control points. Here, higher-order B-spline (i.e., of degree 7) interpolation is employed to suppress variation in curvature of the leader shapes.

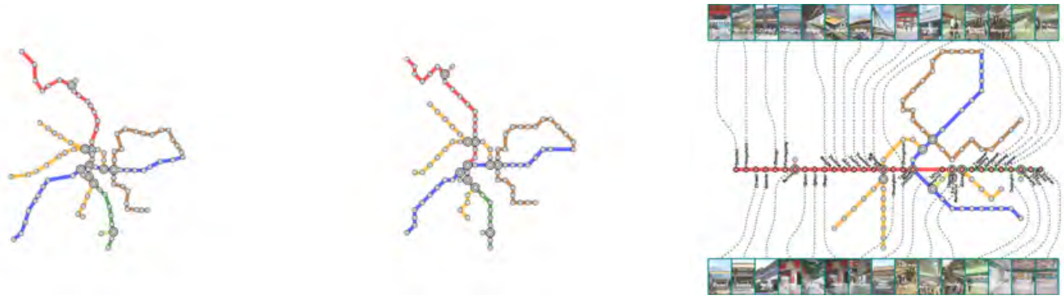
Table 1: *Computation times (in seconds) averaged over 10 trials. The four columns represent the city name, number of stations on the route and entire map, computation times for map layout and leader shapes, respectively.*

Map	On route/Total	Layout	Leaders
Prague	24 / 54	1.85	3.40
Taipei	30 / 89	5.39	6.75
Vienna	26 / 90	7.37	5.83
Munich	15 / 96	7.52	5.35
Singapore	18 / 80	5.32	6.59

5. Results and Evaluation

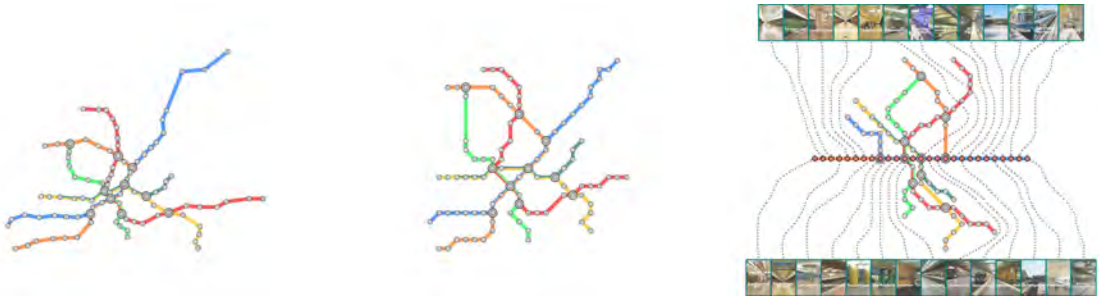
Our prototype system was implemented on a desktop PC with two Quad-Core Intel Xeon CPUs (2.4GHz, 12MB cache) and 8GB RAM, and the source code has been written in C++ using OpenGL for drawing metro lines, OpenCV for handling images, and GLPK (GNU Linear Programming Kit) for solving MIP problems. Table 1 shows computation times required for optimizing map layouts and leader shapes.

Figure 1 presents design examples of the Vienna metro map, including its geographical map, octilinear layout obtained using the conventional MIP formulation, and our travel-route-centered layout. As compared with the conventional octilinear layout in Figure 1(b), our design scenario successfully composes the red-line centered map where the stations close to concert halls and hotels are annotated for guiding EuroVis 2012 participants as exhibited in Figure 1(c). The stations on the route are systematically annotated with thumbnail photographs while textural labels are also aligned vertically at the stations on the same sides of the leaders. Figure 8 corresponds to the case of the Taipei metro map, where the metro network has been transformed into the travel-route-centered layout while local shape features are still preserved. In this example, the route consists of red and green lines, and textural labels are aligned along the leaders emanating from the corresponding stations. Another design example is presented in Figure 9, where the orange and blue lines in the Munich metro have been extended to be horizontally straight along the centerline of the map. As



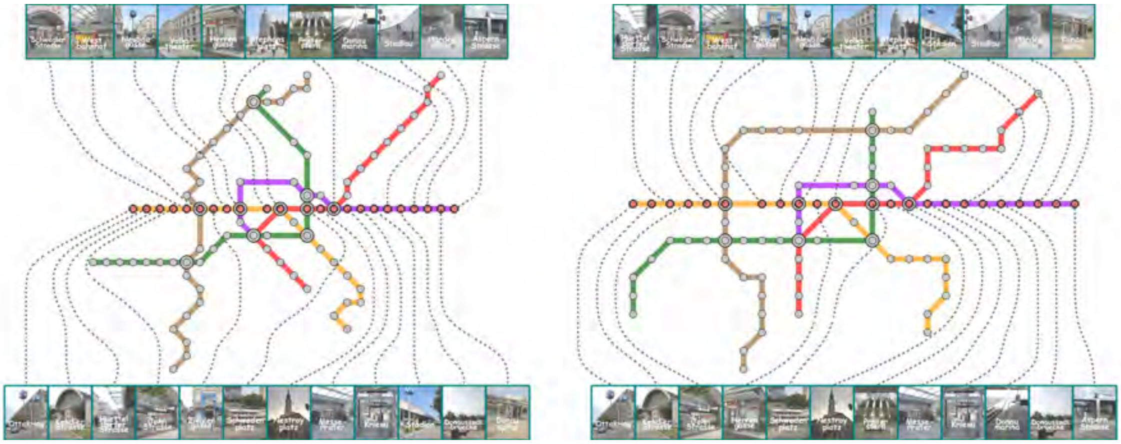
(a) (b) (c)

Figure 8: *Taipei metro map. (a) Geographical layout. (b) Conventional MIP layout. (c) Customized layout using our approach.*



(a) (b) (c)

Figure 9: *Munich metro map. (a) Geographical layout. (b) Conventional MIP layout. (c) Customized layout using our approach.*



(a) (b)

Figure 10: *Interactive design of the Vienna metro map. (a) Original layout. (b) Improved layout through interactive design.*

shown in the figure, the congested downtown area has been automatically rearranged to obtain an aesthetic layout of the metro network, and textural labels are placed inside the labeling boxes in this case. Note that here we can effectively avoid intersections between label leaders and metro lines.

Using our design interface, we can interactively rearrange the travel-route-centered map layout as we like. Figure 10 shows such an example where we try to change the aspect ratio of the original octilinear layout of the Vienna metro network (Figure 10(a)) to the more appropriate one (Figure 10(b)). This example shows the capability of our approach for customizing metro maps through the proposed interface.

In order to confirm that the metro network visually pops up on our customized map layout in practice, we organized two eye-tracking experiments. The first experiment has been

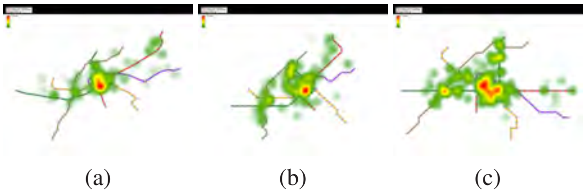

(a) (b) (c)

Figure 11: *Eye-gaze distribution for different metro map layouts. (a) Geographical layout. (b) Conventional MIP layout. (c) Customized layout using our approach.*

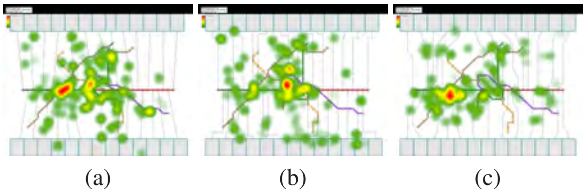

(a) (b) (c)

Figure 12: *Eye-gaze distribution for maps with different leader types. (a) Straight leaders. (b) Orthogonal leaders. (c) Curved leaders ($q = 50$ in Eq. (1)).*

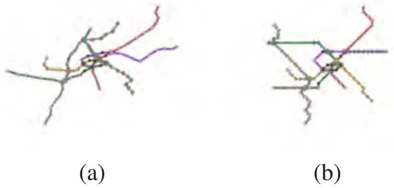

(a) (b)

Figure 13: *Failure cases. (a) The MIP problem with constraints on the route cannot be solved. (b) The route cannot be fully extended to be straight on the map.*

designed to test whether the central route actually directs visual attention from viewers. Here, we invited twelve graduate students working for image synthesis as participants, and asked them to freely view synthesized metro maps on the computer display. Here, each participant viewed metro map images in a random order, where each image is preceded by a dummy image with a white central fixation cross and black background. The participants' eye-gaze movements were tracked using the Tobii X120 eye-tracker. Figure 11 shows eye-gaze distributions on the Vienna metro maps, where Figure 11(a), (b), and (c) correspond to the geographical layout, conventional octilinear layout, and the proposed route-centered layout, respectively. The red-yellow-green color legend represents how long the corresponding position has been viewed between 0.5 to 2.5 seconds after the map image was presented. Figure 11(c) reveals that the elongated route along the map center successfully attracted the longest duration of gaze-fixation. In the second experiment, we investigated how different types of leader shapes influence the visual attention to the map content. We requested the participants to view Vienna metro maps with different types of leader shapes, i.e., straight, orthogonal, and curved leaders with few intersections with metro lines, as shown in Figure 12. According to the distributions of gaze fixation on these maps, the participants were most likely to focus their attention on the travel route when it was annotated with the curved leaders. This implies that intersections between label leaders and metro lines considerably degrade the readability of the map because we cannot instantly discriminate between these two types of line features, especially when they are intricately intersected.

Our approach has several limitations. The existing MIP solver cannot find a travel-route-centered layout when the route contains corner points (Figure 13(a)). The specified route sometimes cannot be fully extended to be straight especially when it is topologically constrained by many crossings with different metro lines (Figure 13(b)). The flow network computation usually provides satisfactory leader shapes and only fails when drastic distortion is intentionally introduced to the specified travel route. It is also important to control the curvature of label leaders for visual quality of the map content, while we leave this untouched as the users can fully edit the leader shapes by adjusting the parameters of the edge weight in Eq. (1).

6. Conclusion

This paper has presented an approach for customizing metro map layouts so that we can annotate stations of the specific travel route with large annotation labels. The idea behind this approach is to elongate the travel route to be horizontally straight and systematically annotate stations on the route using boundary labeling. The conventional MIP formulation has been devised to create route-centered octilinear layouts, where we associate angle variables with metro line segments and edit them to rearrange the shape of the metro network. The shapes of the label leaders are also aesthetically arranged through the flow network computation to avoid unwanted intersections between leaders and metro lines.

Our future extensions include fully evaluating usability of our travel-route-centered layout and annotation in both static and animated metro maps. We should also allow additional shape patterns for the travel routes so as to provide more flexibility in customizing the metro maps. Implementing our technique on mobile devices is also an important future research theme.

Acknowledgements

The authors would like to thank anonymous reviewers for their helpful comments. This work has been partially supported by JSPS under Grants-in-Aid for Scientific Research (B) No. 24330033 and challenging Exploratory Research No. 23650042, and NSC 98-2218-E-009-026-MY3 and NSC 100-2221-E-002-132-MY3, Taiwan.

References

[ABKS10] ARGYRIOU E., BEKOS M. A., KAUFMANN M., SYMVONIS A.: On metro-line crossing minimization. *Journal of Graph Algorithms and Applications 14*, 1 (2010), 75–96. 3

[AS01] AGRAWALA M., STOLTE C.: Rendering effective route maps: Improving usability through generalization. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH 2001, pp. 241–249. 2, 3

[BKNS10] BEKOS M. A., KAUFMANN M., NÖLLENBURG M., SYMVONIS A.: Boundary labeling with octilinear leaders. *Algorithmica 57*, 3 (2010), 436–461. 3

[BKPS08] BEKOS M. A., KAUFMANN M., POTIKA K., SYMVONIS A.: Line crossing minimization on metro maps. In *Proceedings of the 15th International Conference on Graph Drawing* (2008), vol. 4875 of *Springer Lecture Notes in Computer Science*, pp. 231–242. 3

[BKPS11] BEKOS M. A., KAUFMANN M., PAPADOPOULOS D., SYMVONIS A.: Combining traditional map labeling with boundary labeling. In *Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM2011)* (2011), vol. 6543 of *Springer Lecture Notes in Computer Science*, pp. 111–122. 3

[BKSW07] BEKOS M. A., KAUFMANN M., SYMVONIS A., WOLFF A.: Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry: Theory and Applications 36*, 3 (2007), 215–236. 3, 5

[DHM08] DWYER T., HURST N., MERRICK D.: A fast and simple heuristic for metro map path simplification. In *Advances in Visual Computing* (2008), vol. 5359 of *Springer Lecture Notes in Computer Science*, pp. 22–30. 3

[EK72] EDMONDS J., KARP R. M.: Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM 19*, 2 (1972), 248–264. 5, 6

[HMD05] HONG S.-H., MERRICK D., DO NASCIMENTO H. A. D.: The metro map layout problem. In *Proceedings of the 12th International Symposium on Graph Drawing* (2005), vol. 3383 of *Springer Lecture Notes in Computer Science*, pp. 482–491. 2, 3

[KI03] KAMEDA T., IMAI K.: Map label placement for points and curves. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E86-A*, 4 (2003), 835–840. 3

[Lin10] LIN C.-C.: Crossing-free many-to-one boundary labeling with hyperleaders. In *Proceedings of the IEEE Pacific Visualization Symposium 2010* (2010), pp. 185–192. 3

[LKY08] LIN C.-C., KAO H.-J., YEN H.-C.: Many-to-one boundary labeling. *Journal of Graph Algorithm and Appplications 12*, 3 (2008), 319–356. 3

[Mac86] MACEACHREN A. M.: A linear view of the world: Strip maps as a unique form of cartographic representation. *The American Cartographer 13*, 1 (1986), 7–25. 2, 3

[MG06] MERRICK D., GUDMUNDSSON J.: Path simplification for metro map layout. In *Proceedings of the 14th International Symposium on Graph Drwaing* (2006), vol. 4372 of *Springer Lecture Notes in Computer Science*, pp. 258–269. 3

[MJ87] MACEACHREN A. M., JOHNSON G.: The evolution, application and implications of strip format travel maps. *The Cartographic Journal 24*, 2 (1987), 147–158. 2, 3

[NW11] NÖLLENBURG M., WOLFF A.: Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Transactions on Visualization and Computer Graphics 17*, 5 (2011), 626–641. 3, 4

[Rob03] ROBERTS M. J.: *Underground Maps After Beck*. Capital Transport, 2003. 2

[SRMOW11] STOTT J., RODGERS P., MARTINEZ-OVANDO J. C., WALKER S. G.: Automatic metro map layout using multicriteria optimization. *IEEE Transactions on Visualization and Computer Graphics 17*, 1 (2011), 101–114. 2, 3

[SRP10] SCHMID F., RICHTER K.-F., PETERS D.: Route aware maps: Multigranular wayfinding assistance. *Spatial Cognition and Computation 10*, 2 (2010), 184–206. 3

[SSB06] STADLER G., STEINER T., BEIGLBÖCK J.: A practical map labeling algorithm utilizing morphological image processing and force-directed methods. *Cartography and Geographic Information Science 33*, 3 (2006), 207–215. 3

[WC11] WANG Y.-S., CHI M.-T.: Focus+context metro maps. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2528–2535. 3

[Wol07] WOLFF A.: Drawing subway maps: A survey. *Informatik - Forschung und Entwicklung 22*, 1 (2007), 23–44. 2

[WPCM02] WARE C., PURCHASE H., COLPOYS L., MCGILL M.: Cognitive measurements of graph aesthetics. *Information Visualization 1* (2002), 103–110. 1

[WTLY11] WU H.-Y., TAKAHASHI S., LIN C.-C., YEN H.-C.: A zone-based approach for placing annotation labels on metro maps. In *Proceedings of the 11th International Conference on Smart Graphics* (2011), vol. 6815 of *Springer Lecture Notes in Computer Science*, pp. 91–102. 3

[WWKS01] WAGNER F., WOLFF A., KAPOOR V., STRIJK T.: Three rules suffice for good label placement. *Algorithmica 30*, 2 (2001), 334–349. 3