A Zone-Based Approach for Placing Annotation Labels on Metro Maps

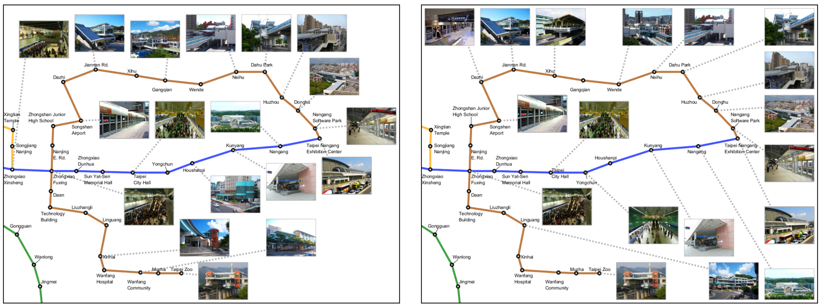
Hsiang-Yun Wu[1], Shigeo Takahashi[1], Chun-Cheng Lin[2], and Hsu-Chun Yen[3]

[1] Dept. of Complexity Science and Engineering, The University of Tokyo, Kashiwa, Chiba, Japan yun@visual.k.u-tokyo.ac.jp, takahashis@acm.org
[2] Dept. of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan cclin321@nctu.edu.tw
[3] Dept. of Electrical Engineering, National Taiwan University, Taipei, Taiwan yen@cc.ee.ntu.edu.tw

Abstract. Hand-drawn metro map illustrations often employ both internal and external labels in a way that they can assign enough information such as textual and image annotations to each landmark. Nonetheless, automatically tailoring the aesthetic layout of both textual and image labels together is still a challenging task, due to the complicated shape of the labeling space available around the metro network. In this paper, we present a zone-based approach for placing such annotation labels so that we can fully enhance the aesthetic criteria of the label arrangement. Our algorithm begins by decomposing the map domain into three different zones where we can limit the position of each label according to its type. The optimal positions of labels of each type are evaluated by referring to the zone segmentation over the map. Finally, a new genetic-based approach is introduced to compute the optimal layout of such annotation labels, where the order in which the labels are embedded into the map is improved through the evolutional computation algorithm. We also equipped a semantic zoom functionality, so that we can freely change the position and scale of the metro map.

1 Introduction

Annotating landmarks with *texts* and *images* is a popular technique for guiding specific travel routes and places of interest, especially in commercially available guide maps. In such map annotation, both internal and external labeling techniques play an important role. *Internal labels* are placed close enough to the reference points called *sites*, and usually used to annotate landmarks with *small* labels containing textual information. On the other hand, *external labels* are often used to assign *large* annotation labels, such as reference thumbnail photographs, and usually spaced sufficiently apart from the corresponding sites, whereas line segments called *leaders* are introduced to connect the sites and labels to clarify their correspondence. Internal labeling instantly allows us to find correspondence between the sites and annotation labels while it usually suffers from the space limitations especially when we have to fit many labels into a



(a) (b)

Fig. 1. Examples of annotation label layouts in Taipei MRT maps: (a) A conventional layout where image labels are placed closer to the corresponding sites. (b) The proposed layout into which commonly used design in hand-drawn maps is incorporated.

small space around the map content. External labeling overcomes this problem by seeking more labeling space away from the map content, but at the cost of applying leaders that often disturb the visual quality of the annotated map.

In hand-drawn metro map illustrations, the mixture of internal and external labels is effectively employed to apply textual and image annotation labels. However, the arrangements of such textual and image labels are quite different in that the textual labels are placed in the vicinity of the corresponding site, while the image labels are more likely to be aligned around the corner of the map domain or along its boundaries, so that we can fully enhance the *aesthetic* arrangement of such labels on the entire map as shown in Fig. 1(b). Indeed, conventional approaches attach image labels in the same way as the textual ones, resulting in a map that cannot maintain a visually plausible arrangement of image labels, as seen in Fig. 1(a). Nonetheless, formulating this kind of aesthetic map design as a computational algorithm is still a challenging task in the sense that we have to solve a rectangle packing problem, which usually leads to a well-known combinatorial NP-hard problem. Furthermore, the problem becomes more complicated especially for annotating metro maps. This because the labeling space around the metro network often consists of multiple non-rectangular regions including small ones, where we cannot directly apply the conventional boundary labeling techniques neither.

In this paper, we present a new approach for placing textual and image annotation labels on the metro maps while maintaining the above aesthetic arrangement of such labels in the map domain. This is accomplished by segmenting the entire map into content, internal, and external zones, so that we can arrange annotation labels according to their label types. We compute such zone-based segmentation by applying conventional image processing techniques to dilate the network of metro lines on the map image. This segmentation also allows us to introduce the potential field over the outermost external zone, where we can aesthetically align the image annotation labels along the map boundaries by referring to the potential values. Note that in our setup, we take a metro network as input in which each vertex of the underlying graph represents a station together with its geographical position and each edge corresponds to the metro route between the corresponding pair of adjacent stations. We also assume that textual and image annotation labels correspond to each station of the metro network, and retain the name of the station and a thumbnail photograph of the view around that station, respectively, as shown in Fig. 1.

We formulate the optimal placement of textual and image annotation labels as a combinatorial optimization and search problem, employing genetic algorithms (GA) as a heuristic optimization method to simulate the process of natural evolution. In the process of solving the problem, each chromosome is defined as a value-encoding sequence of label IDs, where each label is embedded into vacant labeling space in a greedy fashion one by one in order. In our approach, in order to limit the number of possible positions of each label over the map, the labeling space around the metro network is decomposed into a set of square grids in each zone, which effectively reduces the computational complexity for the label placement. The final label layout is obtained by optimizing the function that penalizes the number of missing labels and sum of the normalized leader lengths.

The remainder of this paper is structured as follows: Sect. 2 provides a survey on conventional techniques for label placement and metro map visualization. Sect. 3 describes how we can segment the entire map domain into several zones using image processing techniques. Sect. 4 provides our new genetic-based approach to fitting the textual and image annotation labels within the labeling space aesthetically around the the metro map content. After having presented several experimental results in Sect. 5, we conclude this paper in Sect. 6.

2 Related Work

Annotating point features has been one of the fundamental techniques especially in the area of cartography. Christensen et al. [5] conducted an empirical study on several heuristic algorithms for this purpose. Applying internal labels to point features has been intensively investigated so far, and several schemes for interactive design of label placement [8], real-time label arrangement [16], and consistent label layout in dynamic environments [1] have been developed.

On the other hand, the concept of boundary labeling has been introduced by Bekos et al. [3] as one of the external labeling techniques. They provided mathematical formulations together with efficient algorithms for arranging labels on the boundary margins around the rectangular content area. Lin et al. also performed theoretical studies on several different types of correspondences between the sites and boundary labels including one-to-one correspondence [14], multiple-to-one correspondence [13], and its improved version with hyper-leaders and dummy labels [12].

Employing both internal and external labels provides an effective means of annotating map features while it has still been a challenging research theme. Several functional and aesthetic criteria have been proposed for this purpose by formulating the label layouts in hand-drawn illustrations [10], which were followed by a real-time algorithm for annotating 3D objects [9]. However, these schemes used the external labels as the replacements of internal ones when the labels cannot be placed close enough to the corresponding sites, and always tried to embed both types of labels as close as possible to the sites. (See Fig. 1(a).) Bekos et al. [2] also presented the combined use of internal and external labels while the external labels were expected to stay on the predefined boundary margins only. Our proposed approach differs from the conventional ones in that it also takes advantage of a set of small empty space around the given metro network as the labeling space, while keeping the aforementioned aesthetic layout of textual and image annotation labels on the entire map domain. Combinations of internal and external labels have also been employed to annotate various targets including column charts [17], 3D virtual landscapes [15], surfaces of 3D bumpy objects [6], and 3D illustrations [7].

Metro map visualization itself is an interesting theme that has been intensively researched recently [21]. In this category, the metro map was aesthetically deformed first and the stations were then annotated by textual labels. Hong et al. [11] presented an approach to visualizing metro maps based on the spring models, Böttger et al. [4] developed a scheme for distorting metro maps for annotation purposes, Stott et al. [19] formulated multiple criteria for optimizing metro map layout and placing textual labels and Nöllenburg and Wolff [18] employed mixed-integer programming to draw and label metro maps in a visually plausible manner. In our approach, we respect the original geometry of the given metro network and focus on the optimal placement of annotation labels only.

3 Zone-Based Segmentation of the Map Domain

Our approach begins by partitioning the metro map domain into three zones: the content, internal, and external zones. The *content zone* tightly encloses the metro network and we do not place any labels in that zone. The *internal zone* is next to the content zone and we can place textural labels only there. The *external zone* is the complement of the previous two zones and we can embed any types of labels there. Note that in this zone-base map segmentation, we can lay out textual labels on both the internal and external zones while image labels can be placed within the external zone only, as shown in Fig. 3.

3.1 Dilating Metro Lines on the Map Image

For defining the three different zones over the metro map, we first generate the metro network image by drawing the given metro lines, then apply the morphological dilation operations to the image to synthesize dilated metro network images. Note that, in our implementation, we employed the ordinary 3×3 rectangular kernel mask, which is equipped with the OpenCV library by default. Although Cipriano and Gleicher [6] applied both the dilation and erosion operations to compute the scaffold surface over the input 3D bumpy objects, our

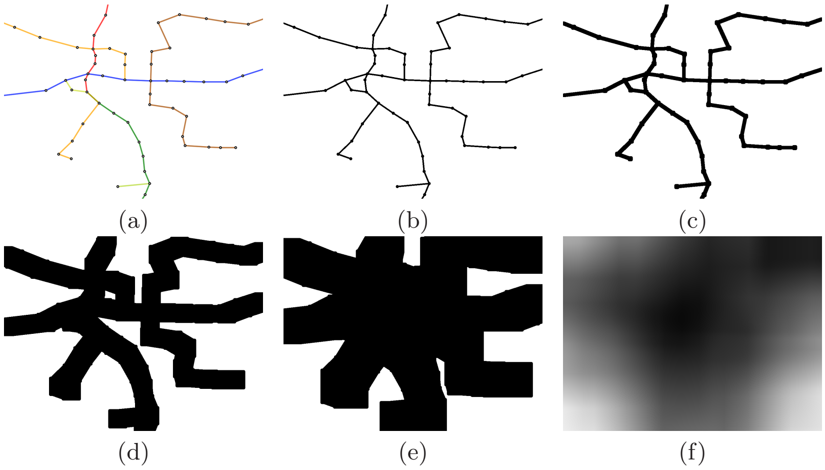


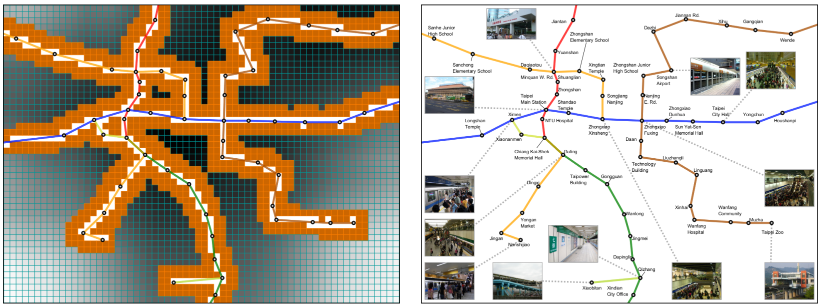
Fig. 2. Partitioning a metro map domain into three zones. (a) Original metro network rendered from the input graph data. (b) Binary metro network image. (c) Content zone obtained by a small number of dilation operations. (d) Internal zone obtained by a medium number of dilation operations. (e) Tentative zone obtained by a large number of dilation operations. (f) Potential field obtained by applying Gaussian filtering to (e). Black and white colors correspond to high and low potential regions, respectively.

approach just uses the dilation operations since our aim here is to design the zone-based segmentation of the labeling space around the metro network.

Fig. 2 shows how an original metro network image is dilated for obtaining the aforementioned three zones of the map domain. Fig. 2(a) represents the original metro network image rendered by taking as input the graph of metro lines. The first task here is to convert this color image into a black-and-white binary image as shown in Fig. 2(b). We are now ready to define the content zone by applying dilation operations n_c times to this binary image and extract the black region from the resulting dilated image in Fig. 2(c). The internal zone can be defined in the same way as the black region in Fig. 2(d) by applying the dilation operations n_i times, where $n_i > n_c$. Finally, the complement of the content and internal zones is defined as the external zone, which corresponds to the white region in Fig. 2(d). Note that, in our implementation, we set $n_c = 4$ and $n_i = 32$ by default in the above image dilation stages.

3.2 Defining Potential Fields for Image Annotation Labels

We also define a potential field specifically for placing image annotation labels so that we can align them along the boundary of the map domain in a greedy fashion for later use. (See Sect. 4.) The potential field that we are going to formulate here is similar to the distance field that Götzelmann et al. [9] used for annotating 3D objects, while it is different in that our potential field is the reversed version of the distance field in order to align image labels along the



(a) (b)

Fig. 3. (a) Zone-based segmentation on Taipei MRT map. White, orange, and gray colors (with gradation) indicate the content, internal, and external zones, respectively. In the external zone, black and white colors indicate high and low potential regions, respectively. (b) Taipei MRT map with textual and image annotation labels.

map boundary rather than in the neighborhood of the corresponding site. We also keep potential values rather uniform in the region away from the boundary to make image labels freely move around in that region to avoid undesirable conflicts with other labels. The potential field has been defined in our approach again by applying dilation operations n_e times, where $n_e > n_i$, to the binary image in Fig. 2(b), so as to obtain the sufficiently dilated metro network image as shown in Fig. 2(e). This dilated image is then blurred with the Gaussian filter to obtain the potential field as shown in Fig. 2(f), where the black and white colors indicate high and low potential regions, respectively. In our implementation, $n_e = 64$ by default.

3.3 Discretizing the Map Domain into Grid Square Cells

Basically, we can find good positions for each label by referring to the zone-based map partition and potential field that we have obtained. However, allowing the annotation labels to move over the map domain pixel by pixel leads to the excessive degree of freedom in their position. In our approach we discretize the map domain into a set of grid square cells in order to effectively limit the number of available positions for each label, which allows us to reduce the search space for optimized label placements. Note that in our implementation, we fixed the side length of the grid square to be 16 pixels. This discrete representation of the map domain can be easily obtained by dividing the side lengths of the image by the grid square size. We then refer to each pixel value of the resized image for retrieving the zone-based map segmentation and potential values. In our implementation, we use the OpenCV library again to perform the image resizing operations. Fig. 3(a) represents an example of the resulting set of grid square cells together with the zone-based segmentation and potential field described previously.

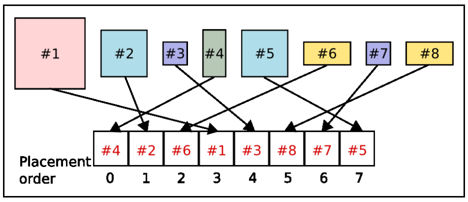


Fig. 4. A chromosome defined as a value-encoding sequence of label IDs. The annotation labels will be fit into the vacant space around the metro network one by one.

4 Genetic-Based Optimization of Label Placement

In this section, we present a new approach for automatically laying out textual and image annotation labels in the space of arbitrary shape around the metro network. Our idea here is to introduce a genetic-based approach that allows us to fit annotation labels effectively into the limited labeling space.

4.1 Encoding the Order of Label IDs

In our genetic-based formulation, each chromosome is defined as a value-encoding sequence of label IDs as shown in Fig. 4, where the annotation labels will be fit into vacant labeling regions in a greedy fashion one by one in the order the label IDs appear in the sequence. We first initialize the chromosome pool by a set of randomly ordered sequences of label IDs, then improve the quality of the chromosome pool by discarding bad chromosomes and reproducing better children from the selected fine parent chromosomes using crossover and mutation operations. Note here that these crossover and mutation operations are carried out while maintaining the condition that each label ID appears only once in each chromosome sequence. Of course, we only consider sites that are contained in the current map domain when we zoom in/out the map content.

4.2 Greedy Placement of Textual and Image Annotation Labels

In our GA-based formulation, the position of each label is uniquely determined by the order the corresponding label IDs in the chromosome. For this purpose, we have to seek the best position of each label while avoiding possible overlaps and crossings with other labels that have already been fixed so far. For placing annotation labels, we try to fit them into a vacant region so that it becomes the closest to the corresponding site. For example, suppose that we have already placed the first three labels of the sequence Fig. 4 step by step, and try to find the optimal position of Label #1 next as shown in Fig. 5. We locate the best position for Label #1 that maximizes the closeness to the corresponding site on the condition that the label can avoid any overlaps and crossings with the existing labels in a greedy fashion.

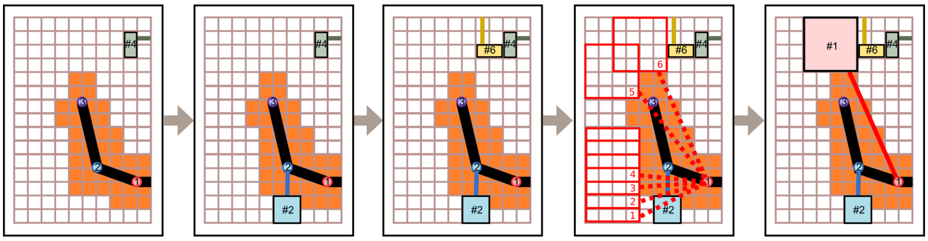
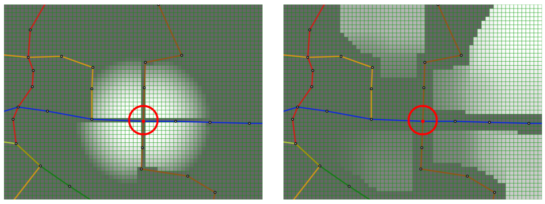


Fig. 5. Greedy search for the best positions of labels in the order of closeness to the sites.



(a) (b)

Fig. 6. Spatial distribution of the fitness values for placing (a) textual and (b) image labels around the transfer station in circles. White color indicates the better positions.

This way, each textual label can be matched with its corresponding site because they are close enough to each other in general. However, if their distance exceeds the predefined threshold, we need to draw a leader between them to fully clarify the correspondence between the station and its name in the metro map visualization. Such a case can be found in Fig. 3(b). For the image annotation labels, on the other hand, we use a different strategy to find the optimal positions. As described earlier, we already synthesized the potential field in the external zone, which allows us to align the image labels around the corner of the map domain or along its boundaries. For finding the best position for an image label, we compute the sum of the potential values on the square cells that will be covered by the label at each possible position, and then employ the position having the optimal sum while avoiding conflicts with other existing labels on the map.

In our framework, we place textual and image annotation labels individually using two different chromosome pools. Actually, we place textual labels first and then image labels, where we try to avoid overlap between the textual and image labels while allowing crossings between the textural labels and leaders connected to the image labels. This is because we can considerably alleviate the visual flickers due to such crossings, by assigning different colors to the texts and leaders of the image annotation labels, as shown in Fig. 3(b).

Moreover, we also design the arrangement of annotation labels so that all the labels are free of conflicts with the metro network itself. In our implementation, we first compute the list of best possible positions for each textual or image label in the preprocessing stage, and then explore the optimal position of the annotation label by visiting the list of possible positions from the head to the tail. (See Fig. 5 also.) Note that we limit the size of the list for each label up to 500 positions in our implementation, and give up placing the label if we cannot find any conflict-free positions in the list. Figs. 6(a) and (b) show the spatial distribution of the position fitness for textual and image annotation labels, respectively, on the grid square cells around the crossing station node in circles. Here, the white color corresponds to the better positions of the labels. We search for the best conflict-free positions of annotation labels in a greedy fashion according to their types, in the order it appears on the corresponding chromosome sequence. As for the leader of each label, we sample the points on the label boundary and employ the one that minimizes the leader length as the joint between the leader and label.

4.3 Definition of an Objective Function

For evaluating the goodness of each chromosome, we use the same objective function both for textual and image annotation labels. In our framework, we count the number of the annotation labels that are blocked out from the labeling space by other existing labels, then increase the penalty score accordingly in order to penalize missing labels. We also compute the total sum of the normalized leader lengths so that we can enhance the visual quality of the label layout by minimizing the associated distances between the labels and sites. The actual definition of our objective function can be written as

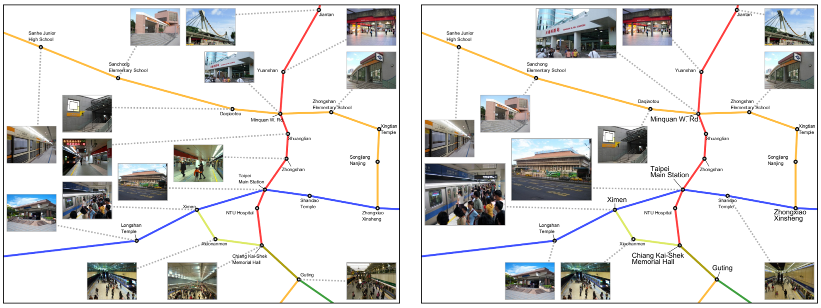
$$\lambda_1 \times f_{\text{penalty}} + \lambda_2 \times f_{\text{leader}},$$

where f_{penalty} and f_{leader} represent the penalties of the missing labels and the sum of the normalized leader lengths, respectively, and λ_1 and λ_2 indicate the corresponding weight values. We set $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ by default in our implementation.

Furthermore, we also assign a priority value to each label in order to make important labels more likely to stay in the metro map. This can be accomplished by just multiplying the penalty scores of each label by the corresponding priority value in the above definition. With this strategy, we can retain the important annotation labels within the labeling space on the finalized metro map.

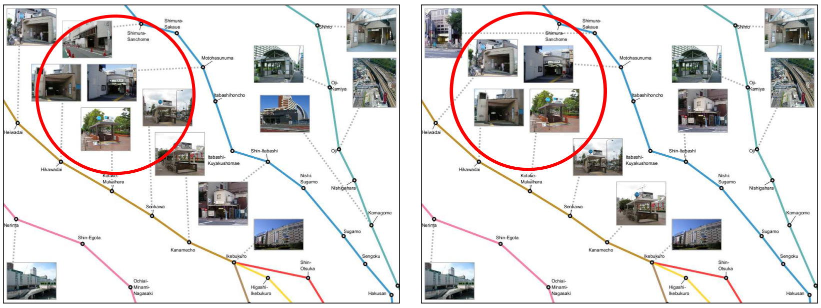
5 Results

Our prototype system has been implemented on a laptop PC with an Intel Core i7 CPU (2.67GHz, 4MB cache) and 8GB RAM, and the source code is written in C++ using the OpenGL library, OpenCV library and GLUT toolkit. The implementation of the genetic-based computation in our system has been based on the GAlib package [20]. Note that all the resulting map images were synthesized at the resolution of 1024×768 in this paper.



(a) (b)

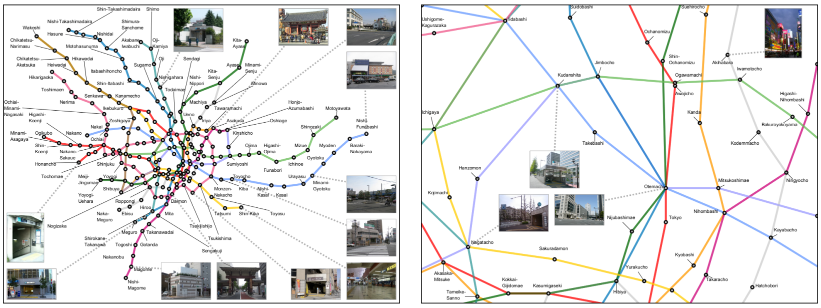
Fig. 7. (a) Uniform and (b) adaptive label size adjustment for Taipei MRT maps. Important stations are emphasized by enlarging the text and image annotation labels.



(a) (b)

Fig. 8. Influence of different weight values λ_1 and λ_2 on the label layouts over Tokyo subway maps. (a) $\lambda_1 = 0.8$ and $\lambda_2 = 0.2$. (b) $\lambda_1 = 0.2$ and $\lambda_2 = 0.8$. The number of labels is maximized in (a) while the total sum of leader lengths is minimized in (b).

Fig. 3(a) shows the zone-based segmentation of Taipei MRT map calculated in our system and Fig. 3(b) presents the finalized layout of textual and image annotation labels on the map. The synthesized map clearly shows that we can aesthetically align image annotation labels around the corner or along the boundaries of the map domain, while textual labels are placed close enough to the corresponding stations. Note here that we assign higher priority to transfer stations and terminals by default in our system, thus annotation labels are more likely to be applied to these stations. We can also emphasize such important stations explicitly by enlarging the corresponding annotation labels as demonstrated in Fig. 7(b), which successfully draw more attention to some specific stations compared to Fig. 7(a). Moreover, the label layout can be controlled by tweaking the weight values in the objective function for our genetic-based optimization. Fig. 8 exposes such an example where we can increase the number of embeddable labels (Fig. 8(a)) or minimize the total leader lengths (Fig. 8(b)). Fig. 9 demonstrates that our system also provides a semantic zoom interface for



(a) (b)

Fig. 9. Semantic zoom visualization. Tokyo subway maps at (a) the original scale and (b) finer scale.

Tokyo subway maps. This allows us to inspect an interesting region in more detail without missing the global context by freely changing the position and scale of the map. The computation cost for the label placement depends on the number of stations in the window and its distribution, while our interface basically provide interactive responses.

6 Conclusion

This paper has presented an approach for automatically designing the aesthetic layout of textual and image annotation labels to embed supplemental information into the metro map. The metro map domain is first partitioned into three zones so that we can systematically lay out the textual and image labels by referring to the aesthetic criteria induced from hand-drawn guide maps. Our encoding of label placement was implemented using genetic algorithms to find optimized layout of both types of labels in an interactive environment. Possible future extension includes persistent placement of important labels across multiple scales for visually plausible map visualization. Optimizing the layout of both the annotation labels and metro networks will be also an interesting research theme. A more sophisticated interface for exploring the annotated metro map content with a variety of labeling box and leader styles remains to be implemented.

Acknowledgements: This work has been partially supported by JSPS under Grants-in-Aid for Scientific Research (B) (20300033 and 21300033), NSC 98-2218-E-009-026-MY3, Taiwan, and NSC 97-2221-E-002-094-MY3, Taiwan.

References

1. Been, K., Nöllenburg, M., Poon, S.H., Wolff, A.: Optimizing active ranges for consistent dynamic map labeling. Computational Geometry: Theory and Applications 43(3), 312–328 (2010)

2. Bekos, M.A., Kaufmann, M., Papadopoulos, D., Symvonis, A.: Combining traditional map labeling with boundary labeling. In: Proc. 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM2011). Springer LNCS, vol. 6543, pp. 111–122 (2011)
3. Bekos, M.A., Kaufmann, M., Symvonis, A., Wolff, A.: Boundary labeling: Models and efficient algorithms for rectangular maps. Computational Geometry: Theory and Applications 36, 215–236 (2007)
4. Böttger, J., Brandes, U., Deussen, O., Ziezold, H.: Map warping for the annotation of metro maps. IEEE Comptuer Graphics and Applications 28(5), 56–65 (2008)
5. Christensen, J., Marks, J., Shieber, S.: An empirical study of algorithms for point-feature label placement. ACM Trans. Graphics 14(3), 203–232 (1995)
6. Cipriano, G., Gleicher, M.: Text scaffolds for effective surface labeling. IEEE Trans. Visualization and Computer Graphics 14(6), 1675–1682 (2008)
7. Čmolík, L., Bittner, J.: Layout-aware optimization for interactive labeling of 3D models. Computers and Graphics 34, 378–387 (2010)
8. do Nascimento, H.A.D., Eades, P.: User hints for map labeling. Journal of Visual Languages and Computing 19, 39–74 (2008)
9. Götzelmann, T., Hartmann, K., Strothotte, T.: Agent-based annotation of interactive 3D visualization. In: Proc. 6th International Symposium on Smart Graphics (SG2006). Springer LNCS, vol. 4073, pp. 24–35 (2006)
10. Hartmann, K., Gotzelmann, T., Ali, K., Strothotte, T.: Metrics for functional and aesthetic label layouts. In: Proc. 5th International Symposium on Smart Graphics (SG2005). Springer LNCS, vol. 3638, pp. 115–126 (2005)
11. Hong, S.H., Merrick, D., do Nascimento, H.A.D.: Automatic visualisation of metro maps. Journal of Visual Language and Computing 17, 203–224 (2006)
12. Lin, C.C.: Crossing-free many-to-one boundary labeling with hyperleaders. In: Proc. IEEE Pacific Visualization Symposium 2010 (PacificVis 2010). pp. 185–192 (2010)
13. Lin, C.C., Kao, H.J., Yen, H.C.: Many-to-one boundary labeling. Journal of Graph Algorithm and Appplications 12(3), 319–356 (2008)
14. Lin, C.C., Wu, H.Y., Yen, H.C.: Boundary labeling in text annotation. In: Proc. 13th International Conference on Information Visualization (IV2009). pp. 110–115 (2009)
15. Maass, S., Döllner, J.: Efficient view management for dynamic annotation placement in virtual landscapes. In: Proc. 6th International Symposium on Smart Graphics (SG2006). Springer LNCS, vol. 4073, pp. 1–12 (2006)
16. Mote, K.: Fast point-feature label placement for dynamic visualizations. Information Visualization 6, 249–260 (2007)
17. Müller, S., Schödl, A.: A smart algorithm for column chart labeling. In: Proc. International Symposium on Smart Graphics (SG2005). Springer LNCS, vol. 3638, pp. 127–137 (2005)
18. Nöllenburg, M., Wolff, A.: Drawing and labeling high-quality metro maps by mixed-integer programming. IEEE Transactions on Visualization and Computer Graphics 17(5), 626–641 (2011)
19. Stott, J., Rodgers, P., Martínez-Ovando, J.C., Walker, S.G.: Automatic metro map layout using multicriteria optimization. IEEE Transactions on Visualization and Computer Graphics 17(1), 101–114 (2011)
20. Wall, M.: GAlib: A C++ library of genetic algorithm components, http://lancet.mit.edu/ga/
21. Wolff, A.: Drawing subway maps: A survey. Informatik - Forschung und Entwicklung 22, 23–44 (2007)