





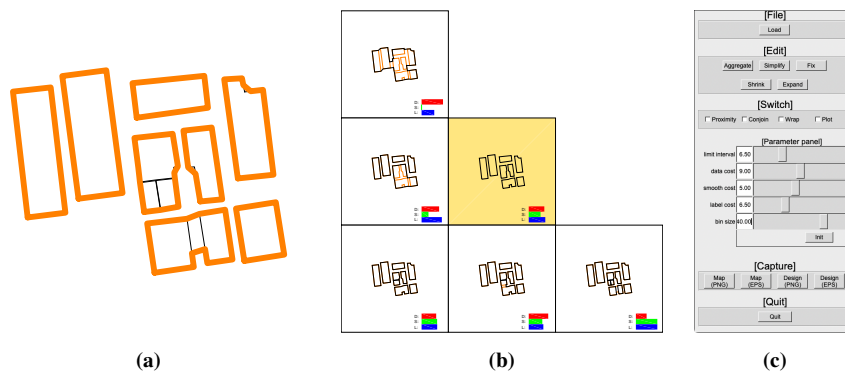
# Interactive Optimization for Cartographic Aggregation of Building Features

Shigeo Takahashi<sup>1</sup> , Ryo Kokubun<sup>1</sup>, Satoshi Nishimura<sup>1</sup> , Kazuo Misue<sup>2</sup> , and Masatoshi Arikawa<sup>3</sup> 

<sup>1</sup>University of Aizu, Japan

<sup>2</sup>University of Tsukuba, Japan

<sup>3</sup>Akita University, Japan



**Figure 1:** A snapshot of the prototype system. (a) Building features in a city block. (b) Candidates for aggregation of building features. (c) A panel for tweaking design parameters.

## Abstract

Aggregation, as an operation of cartographic generalization, provides an effective means of abstracting the configuration of building features by combining them according to the scale reduction of the 2D map. Automating this design process effectively helps professional cartographers design both paper and digital maps, but finding the best aggregation result from the numerous combinations of building features has been a challenge. This paper presents a novel approach to assist cartographers in interactively designing the aggregation of building features in scale-aware map visualization. Our contribution is to provide an appropriate set of candidates for the cartographer to choose from among a limited number of possible combinations of building features. This is achieved by collecting locally optimal solutions that emerge in the course of aggregation operations, formulated as a label cost optimization problem. Users can also explore better aggregation results by interactively adjusting the design parameters to update the set of possible combinations, along with an operator to force the combination of manually selected building features. Each cluster of aggregated building features is tightly enclosed by a concave hull, which is later adaptively simplified to abstract its boundary shapes. Experimental design examples and evaluations by expert cartographers demonstrate the feasibility of the proposed approach to interactive aggregation.

## CCS Concepts

• **Human-centered computing** → **Geographic visualization**; **User interface design**;

## 1. Introduction

Cartographic generalization is a set of methods for exaggerating and suppressing the shapes of geographic features by transforming large-scale maps into small-scale ones. Among them, cartographic aggregation combines small features into large ones to reduce the

complexity of spatial layouts of geographic features in large-scale maps. This powerful tool for editing 2D maps of different scales effectively improves the readability of the entire geographic configuration. It is considered one of the macroscopic generalization operations, in that its design often respects the spatial arrangement

of the overall map features compared to other operations, such as displacement and selection.

Automating aggregation methods significantly reduces the cartographer's workload when designing scale-aware maps. Therefore, optimizing the aggregation of geographic features has particularly exploited the Gestalt principles, which describes human perception for recognizing objects in a group. For the aggregation of small geographic features such as buildings, Gestalt rules have served as standard criteria for exploring the optimal clustering of such map components. However, due to the combinatorial explosion problem, it has been difficult to identify a unique solution among the possible combinations that satisfy the generalization rules. Furthermore, it is almost impossible to satisfy the specific requirements of individual map designers with a single solution, because each designer has his or her own policies and preferences in map design choices. Ultimately, manually specifying the features to be combined became a simple solution to the map design problem, resulting in tedious, time-consuming tasks.

One option for addressing these issues is to provide a set of acceptable solutions for aggregating geographic features from which cartographers can choose their preferred design choices. Alternatively, interactive map designs of such set of solutions could serve a wide range of cartographers with different preferences. Such strategy will greatly reduce the tedious manual selection of geographic features to aggregate if we can effectively filter out unwanted choices. This idea can be further enhanced if the design system has an interface for adjusting associated design parameters to explore different aggregation options.

This paper presents a novel approach to provide such a set of reasonable options in the aggregation of building features. Our key idea is to collect locally optimal solutions in the course of optimizing aggregations of such geographic features. This is achieved by formulating the building aggregation as a label cost optimization problem, that is, as an extension of the graph-cut algorithm for image segmentation.

In this formulation, the aggregation of the respective features is evaluated as the sum of data, smoothness, and label costs, with the definitions of the smoothness and label costs modified from a conventional Gestalt-based conjoining approach [NSX\*11] to fit the cartographic context. The set of aggregated building features is then tightly enclosed by a concave hull to better avoid unwanted mutual conflicts and wastage of surrounding space. The generated hulls are simplified to further reduce the complexity of the boundary shapes, again generating several possible options for selection.

Figure 1 shows a snapshot of the prototype system to demonstrate the feasibility of the proposed approach. The system takes as input a set of building polygons contained in a selected city block (Figure 1a). This returns a set of potential options for aggregating building features (Figure 1b). Users can choose their preferred set of aggregations from the set of building combinations or explore a different set of options by adjusting the design parameters, including the proximity distance threshold, the upper bounds on data, smoothness, and label cost, and the bin size in the vertical histogram of design options with respect to cost (Figure 1c).

The contribution of this paper can be summarized as follows:

- We develop an approach to suggest reasonable design choices for building feature aggregation.
- We propose a set of design parameters to interactively explore different sets of candidates for building aggregation.
- We employ an algorithm for producing concave hulls [MS07] to tightly enclose the aggregated set of features, which is further simplified to reduce shape complexity.

The rest of this paper is organized as follows. Section 2 reviews approaches to cartographic aggregation and other relevant generalization techniques. Section 3 gives an overview of our approach for aggregating building features across different scales. Section 4 describes our idea for collecting reasonable candidates for building aggregations, along with the associated label cost optimization formulation. Section 5 details our algorithm for generating a contour enclosing a cluster of aggregated building features and its simplification process. We present an expected scenario for building aggregation, design examples, an evaluation of the approach, and discussion in Section 6. Section 7 concludes this paper.

## 2. Related Work

We briefly review previous computational approaches to cartographic aggregation and other relevant generalization techniques.

### 2.1. Aggregation as cartographic generalization

The formulation of map generalization originates from the manual design rules of cartographers when transforming a large scale map into a smaller scale map [SM89, Lee96, RMM\*06, MS92, Li06]. However, recent advances in digital map technology have enabled continuous changes in map scale, even on mobile display devices. Thus, semi-automation of map generalization tasks requires computational algorithms that reduce the workload of manual design by cartographers. Among cartographic generalization techniques, aggregation is unique in that it changes the topological connectivity of geographic components in the map by combining multiple features into one. However, properly grouping such geographic features is technically challenging due to the enormous number of possible combinations. Thus, aggregation must be the most involved because of the need to infer the underlying grouping of geographic features and remains to be further investigated.

Regarding exploring the combinations of 2D building features, the Gestalt principles have been widely used because they reasonably reflect the human perception of object grouping. Representative principles of such perceptual grouping include the law of proximity, which tends to group things that are spatially close together, and the law of similarity, which tends to group things that are of the same kind. Li et al. [LYAC04] introduced the combination of the Gestalt theory and mathematical morphology to solve aggregation problems. Liqiang et al. [LHDZ13] successfully formulated the aggregation of geometric features based on Gestalt clustering rules. Pilehforooshha and Karimi [PK19] developed a framework for extracting building linear patterns to define aggregation rules based on their similarity.

In different contexts, such as graphics applications, Nan et

al. [NSX\*11] developed an approach to abstract line drawings, incorporating the Gestalt principles. Based on this approach, we construct a new interactive scheme for aggregating building features. To our knowledge, previous studies on cartographic aggregation have only addressed the problem of categorizing building features into meaningful groups. That is, they have not considered how to compose the profile shapes of aggregated building clusters and have only used conventional off-the-shelf software (such as ArcGIS) to do so. Instead, our approach also considers the contour shapes enclosing the aggregated buildings and incorporates the results into the formulation of aggregation operations.

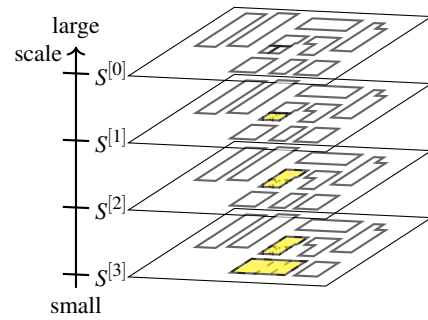
## 2.2. Other cartographic generalization techniques

Cartographic generalization consists of several design operations, each of which maintains the readability of maps with different approaches as their scale changes. Representative operations include displacement, selection, and simplification.

Displacement has been intensively studied in the history of cartography and its related fields, since it directly avoids conflicts between adjacent geometric elements by rearranging their spatial positions. A pioneering approach was offered by Ruas [Rua98], who introduced a Delaunay triangulation to fix the relative positions of geographic elements. Loneragan and Jones [LJ01] formulated the minimum distance between geographic features to maintain the high readability of maps. Since then, the elimination of collisions between geographic elements has been tackled by taking advantage of energy minimization approaches, for example, with simulated annealing [WJT03], genetic algorithms [WWW03], elastic beam trusses [BBW05, LGSM14], and linear programming [MTW\*19].

Selection, also called typification, is introduced to omit small geographic features for reduced map complexity while maintaining visual quality. Thus, this operation has often been used with displacement to eliminate conflicts between geometric components in a small space. Several technical challenges have been met for this purpose using proximity graphs [Reg01], mesh simplification techniques [BC07], optimization based on genetic algorithms [WGL\*17], and Gestalt grouping principles [GW18]. Selection operations have also been successfully incorporated to the automatic composition of schematic route maps with high readability [AS01, Sch08, KAB\*10].

Simplification reduces the complexity of geographic features by eliminating local shape details as the map scale gets smaller. For example, the shapes of geometric entities, such as buildings, are also abstracted by level-of-detail simplification based on image-processing techniques, including the scale-space theory [May05] and a set of optimization techniques [Ses05]. The schematization of line segments has been popular in that many technical challenges have been tackled for rectangular cartograms [SvKF06, BMS11b, BMS11a] and curved schematization [vGMR\*13] along with area preservation [vGMSW14, vGMSW15]. Line schematization has also been addressed by ontology-based semantic analysis [KDE05] and categorization of geometric features [PY11]. Schematization techniques have also been applied to various map styles, such as metro maps [NW11, WC11, WTLY12, WTH\*13], road networks [HS11, CvDH14], and urban area maps [GASP08,



**Figure 2:** Aggregation of building features in the scale-aware map defined by the map domain and scale axis. Aggregated sets of building polygons are colored in yellow.

QWC\*09, HWAT13]. Map schematization was also employed to attract visual attention, with improved map readability [vDvGH\*13].

Recently, many researchers are introducing machine learning techniques to learn the design skills of cartographic generalization by transforming vector map data into raster images, even for aggregation [FTS19, YAY\*22], selection [SLW\*22, XAY\*23], and simplification [YYY\*22].

## 3. Overview

This section describes the results of our survey on how aggregation operations are applied in conventional paper maps. We then summarize the map representation from large to small scales and the design scenarios in our approach.

### 3.1. Investigation of conventional paper maps

We collected old paper maps in Japan and compared them at different scales (1:2,500 vs. 1:10,000 or smaller scales) to investigate how cartographers aggregated building features. Our observation suggests that the building features are likely to be aggregated if (A1) their boundaries are sufficiently close to each other, (A2) their combination forms a simple shape, and (A3) their spatial density becomes low. In addition, the boundaries of aggregated features are simplified so that (S1) they generally become rectangular shapes or (S2) shapes with a small number of corners if they have certain irregularities. See Appendix A in the supplementary material.

As described earlier, we implemented these guidelines by sophisticating a previous approach [NSX\*11], specifically with respect to the proximity of geographic features. Thus, we attempted to aggressively aggregate building features using the proximity rules to respect the above guideline, as well as the similarity rules that were derived from the Gestalt principles in [NSX\*11].

### 3.2. Visualizing maps at different scales

We assumed that we could smoothly zoom in/out the digital representation of maps across different scales, so that we could navigate to the destination at arbitrary scales with mobile devices. In our approach, we used a multi-layered representation of such scale-dependent maps (Figure 2). Specifically, we introduced four layers

at discrete scale samples in our implementation, so that map designers can adaptively aggregate building features at specific scale samples. In Figure 2, we could design different aggregation patterns so that building features gradually merge to reduce complexity as the scale decreases. See Figure 11 for an actual example. Of course, we can add more layers by densely sampling the scale axis to support variable level of detail.

### 3.3. Design scenario

In our approach, we envisioned the following design scenario. A cartographer selects a city block at a given scale in the prototype system by clicking on it with the mouse. The building features in the selected city block are transformed into a set of polygons for later aggregation procedures. The cartographer then asks the system to compute candidate aggregation patterns of the building features and explore the best choice by adjusting the design parameters. If the desired aggregation pattern is not found, the cartographer can force the aggregation of a specific group of buildings by specifying them through the interface. After selecting the aggregated representation of building features from the candidates, the cartographer applies simplification operations to the resulting polygon shapes to design the final aggregated shapes of building features. Finally, the system replaces the original building features in the selected city block with the finalized aggregation patterns.

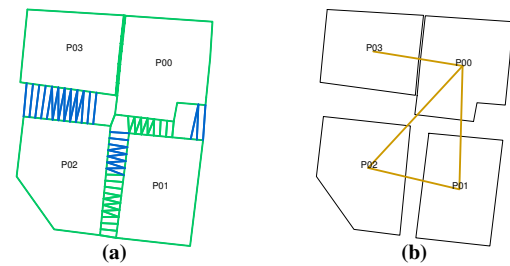
With the multi-layered representation of a scale-aware map, we can design a progressive aggregation of building features in each block. Suppose that we aggregated building polygons at scale  $S^{[1]}$  in a selected city block (Figure 2). In this case, the system first replaces the initial set of building features with aggregated ones at that scale and supersedes building features at smaller scales such as  $S^{[2]}$  and  $S^{[3]}$  with the same aggregated patterns. This helps cartographers to further combine the just obtained aggregation patterns as the map scale decreases. Figure 11 provides such a case in which we designed the progressive aggregation of building features.

## 4. Interactive Aggregation of Building Features

This section describes our approach to collecting acceptable candidates for aggregated patterns of building features. This is accomplished by extending Nan et al.'s [NSX\*11] approach as described earlier. Although this approach employs the Gestalt principles to group components in the same way as conventional aggregation methods, we needed to replace several formulations specifically for proximity-based grouping rules in the context of cartographic aggregation. In addition, we addressed the unique technical challenge of collecting a set of feasible aggregation patterns for building features. Note that we used a city block containing only four building features block (Figure 3) as an example. We also demonstrated visual examples for the city block with complex building layout (Figure 1) in the supplementary material (Appendices B and C).

### 4.1. Composing the proximity graph

Our first step was to construct a proximity graph over the building features. In this case, we considered building polygons as nodes and explore their reasonable proximity for better aggregation operations. The previous approach [NSX\*11] computed the proximity



**Figure 3:** Proximity graph composition. (a) A  $\beta$ -skeleton for evaluating the distance between building features and (b) its associated proximity graph. PXX represents the building ID XX.

between two polygons as the minimum of the maximum distances between samples of the two polygons. However, this does not faithfully follow the proximity rules in our guideline. As described in Section 3.1, three proximity rules derived from our study of conventional paper maps revealed the importance of correctly estimating the distance between building features. This is also justified by the fact that cartographers are unlikely to connect two building features if they identify walking paths between them. We avoided this problem by initially taking dense samples on each edge of the building polygons and composing a  $\beta$ -skeleton over these samples.

We computed the lengths of  $\beta$ -skeleton edges bridging between two different polygons for this purpose. Figure 3 shows the  $\beta$ -skeleton and the proximity graph over the building features in the example city block. Here, green edges correspond to intervals closer than the threshold value, and blue edges correspond to distances above the threshold value. Note that we used  $\beta = 1.2$  for the  $\beta$ -skeletons because this choice is known to effectively produce paths with their centerlines between buildings [RF99].

### 4.2. Label assignment to possible aggregation clusters

Using the proximity graph just constructed, we identified clusters of building features that should be grouped together and assigned a unique label to each of the groups. We introduced previous approach [NSX\*11], and employed the laws of proximity and similarity from possible Gestalt principles. For the law of proximity, however, we improved the techniques for labeling geographic features by computing multi-level proximity graphs.

Suppose that we extract candidate groups of building features by referring to the initial proximity graph. We could extract a list of connected components from the proximity graph and obtain a set of building polygons contained in each component as the group of geographic features to be aggregated. We would then make the proximity graph sparser by slightly reducing the threshold interval (e.g., by 10% of the initial threshold) and pruning the edges if the associated building intervals exceed the threshold. We then retrieve the collection of connected components as a new set of candidate feature groups to aggregate. We repeat this process until the threshold vanishes to collect a sufficient number of candidate groups.

Regarding the law of similarity, we followed the formulation

**Table 1:** Group labels assigned to the buildings in the example city block. PXX indicates the building ID XX, and LXX indicates the label ID XX. Each check mark indicates that the corresponding buildings (PXX) are contained in a specific group label (LXX).

	L00	L01	L02	L03	L04	L05	L06	L07
P00	✓				✓	✓	✓	✓
P01		✓			✓			✓
P02			✓		✓	✓		✓
P03				✓	✓	✓	✓	

of the conventional method [NSX\*11]. Here, we separately transformed the initial proximity graph into a similarity graph by filtering edges if they correspond to pairs of buildings that differ significantly in aspect ratio. Again, we identified the connected components as possible feature groups also for the similarity.

In our formulation, we prepared a group label that contains a single building feature individually—that is, the same number of group labels as the number of building features, which allows us to make each building feature isolated from other features by default. As described above, we also introduced additional group labels to encourage their aggregation (Table 1). Here, L00, L01, L02, and L03 correspond to the set of group labels prepared by default. L04, L05, and L06 are group labels obtained by contracting the proximity graph, while L07 comes from the law of similarity. See Appendix B in the supplementary material also.

#### 4.3. Label cost optimization

When a given geographic feature has multiple group label assignments, we need to determine which group the feature should belong to. We solved this assignment problem using label cost optimization, as used in [NSX\*11]. Label cost optimization [DOIB10, DOIB12] was formulated as an extension of graph cut optimization [BVZ01], which allowed us to effectively solve region segmentation problems in image processing. Graph cut optimization has two types of costs: data cost and smoothness cost. The data cost arises when a feature (e.g., a pixel) is contained in a particular group (e.g., a region), and the smoothness cost is penalized when neighboring features (i.e., pixels) are separated into different groups (e.g., regions). Label cost optimization additionally seeks the optimal grouping of features by newly incorporating the label cost, which penalizes the presence of groups.

Thus, the energy function for the label cost optimization problem was defined with respect to the joint labeling  $f$  and consists of the data cost, the smoothness cost, and the label cost, as follows:

$$E(f) = \sum_{p \in \mathcal{P}} D(p, f_p) + \sum_{p, q \in \mathcal{N}} V_{pq}(f_p, f_q) + \sum_{l \in \mathcal{L}} h_l \cdot \delta_l(f), \quad (1)$$

where  $p \in \mathcal{P}$  represents a feature (i.e., a building polygon), and  $f_p \in \mathcal{L}$  indicates a group label assigned to  $p$ . Thus,  $\mathcal{P} = \{P00, \dots\}$  and  $\mathcal{L} = \{L00, \dots\}$  are sets of features and group labels, respectively. The first term on the right hand side is the data cost, which is the sum of the cost for each feature  $p$  to belong to the label  $f_p$ . The second term is the smoothness cost, which is the sum of the costs  $V_{pq}$  incurred when two features  $p$  and  $q$ , which are in the adjacency

relation  $\mathcal{N}$  in the proximity graph, belong to different group labels  $f_p$  and  $f_q$ . The third term is the label cost, where  $h_l$  indicates the cost for the label  $l \in \mathcal{L}$  and  $\delta_l(f)$  is defined as follows:

$$\delta_l(f) = \begin{cases} 1 & \exists p : f_p = l \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

This means that the cost  $h_l$  associated with the group label  $l$  is accumulated when the group label is assigned to at least one of the geographic features. Refer to [DOIB10, DOIB12] for details.

#### 4.4. Defining the costs

We needed to properly formulate the three types of costs in the label cost optimization problem. For the data cost  $D(p, f_p)$ , we simply adopted the previous formulation used in [NSX\*11]. Thus, we computed the minimum distance between the feature  $p$  and the set of features tagged with the group label  $f_p$ . This is because two features may not be adjacent in the spatial layout.

Further, we replaced the definition of the smoothness cost specifically in our approach because we wanted to retrieve the distance between the two adjacent features that we computed earlier. The smoothness cost between features  $p$  and  $q$  is defined as the inverse of the distance between  $p$  and  $q$ , with the distance defined as the minimum length of the  $\beta$ -skeleton edges connecting  $p$  and  $q$ . This modification faithfully aggregated the building polygons almost in contact, otherwise the cost definition would produce unwanted artifacts in the actual aggregation process (Figure 10a).

We also revised the definition of the label cost for the groups derived from the proximity rules. The previous approach [NSX\*11] formulated the label cost  $h_l$  for the label  $l \in \mathcal{L}$  as the difference between the area of the convex hull covering the group of features and the sum of the areas of the individual features. However, enclosing each group with a convex hull introduced unnecessary space into the hull and created conflicts with surrounding features. This has a significant negative impact on the quality of the label cost optimization and is an obstacle to the desired aggregation operation. Our solution was to replace the convex hull with a so-called concave hull, which encloses a group of building polygons more tightly in a geometric sense (Section 5).

Table 2 lists the data, smoothness, and label costs for four building features (P00–P03) and eight group labels (L00–L07) for the example city block. Notice that we only refer to the smoothness cost associated with two features that are connected with an edge in the proximity graph. Thus, we only show relevant entries (i.e., (P00,P01), (P00,P02), (P00,P03), and (P01,P02)) in the upper triangular part of the symmetric table for the smoothness cost (Table 2b). For the first four group labels that individually cover the building features, we set all label costs to 1.0. We also assigned 1.0 to each feature as the data cost for its own individual group label, while other data costs were set to the upper bound of the data costs. This cost setup prevented each feature from being unintentionally included in other group labels that individually contain different features. Other data, smoothness, and label costs were scaled linearly between the lower and upper bounds. Here, the lower bounds for the data and label costs, excluding the first initial set of individual group labels, were set to 2.0. The lower bounds for the smoothness cost are 1.0 since the cost is irrelevant to the group labels. The

**Table 2:** Costs for the building features in the example city block with four building features. (a) Data (D), (b) smoothness (S), and (c) label (L) costs. The color boxes show correspondence with the costs in Figure 4.

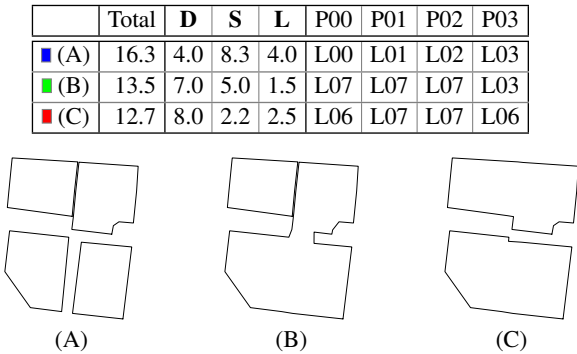
D	L00	L01	L02	L03	L04	L05	L06	L07
P00	1.0 <sup>■</sup>	12.0	12.0	12.0	2.0	2.0	2.0 <sup>■</sup>	2.0 <sup>■</sup>
P01	12.0	1.0 <sup>■</sup>	12.0	12.0	2.0	5.4	12.0	2.0 <sup>■</sup>
P02	12.0	12.0	1.0 <sup>■</sup>	12.0	2.0	2.0	11.9	2.0 <sup>■</sup>
P03	12.0	12.0	12.0	1.0 <sup>■</sup>	2.0	2.0	2.0 <sup>■</sup>	12.0

(a)

S	P00	P01	P02	P03
P00		1.1 <sup>■</sup>	1.1 <sup>■</sup>	5.0 <sup>■</sup>
P01			1.1 <sup>■</sup>	
P02				
P03				

(b)

L	L00	L01	L02	L03	L04	L05	L06	L07
	1.0 <sup>■</sup>	1.0 <sup>■</sup>	1.0 <sup>■</sup>	1.0 <sup>■</sup>	6.7	10.0	2.0 <sup>■</sup>	0.5 <sup>■</sup>

(c)


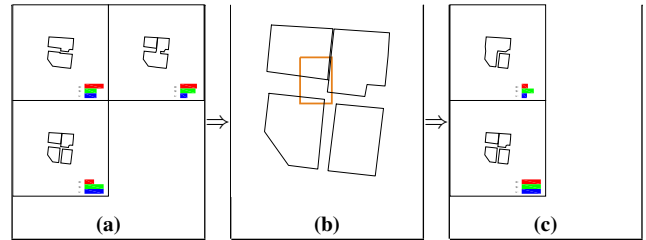
**Figure 4:** Changes in the label assignment  $f_p(\in \mathcal{L})$  to the 11 building features  $p(\mathcal{P})$  for the example city block. Here, D, S, and L denote the data cost, smoothness cost, and label cost, respectively. Note that the total energy decreases as the optimization proceeds.

upper bounds for data, smoothness, and label costs are 12.0, 5.0, and 10.0, respectively, while they will be adjusted interactively.

#### 4.5. Collecting locally optimal solutions

Here, we consider how to collect acceptable candidates for building aggregations by solving the label cost optimization problem. Our idea was to collect locally optimal solutions as the optimization proceeds. This is possible because the  $\alpha$ -expansion algorithm for solving the label cost optimization iteratively replaces the group labels for a set of features until it reaches the optimal solution [DOIB10, DOIB12]. Figure 4 shows how the label assignment changes, where we started from the original layout (A), reduced the total energy by replacing labels (B), and arrived at the global minimum (C). The supplementary material shows locally optimal aggregations for the complex city block (Appendix C).

We also employed color tags to devise the presentation of Figure 4 and Table 2 so that the changes in the data, smoothness, and label costs can be tracked during the optimization process. Here, blue tags correspond to the cost we summed up for the initial layout of building features (A), green tabs for the intermediate layout (B), and red tags for the finally optimized layout (C). For example, we would obtain the smoothness cost for (C) as  $1.1 + 1.1 (= 2.2)$  since these two values are tagged by red in Table 2b. We could also check changes in the costs for the complex city block in the supplementary material (Appendix C).

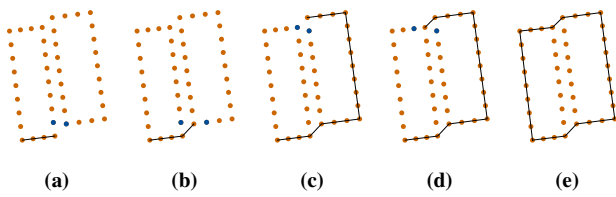


**Figure 5:** An operation to force manually selected features to be aggregated. (a) The initial set of aggregation patterns. (b) Three features in the upper left are selected as a group using a rubber-band interface. (c) The update set of aggregation patterns.

#### 4.6. Finalizing aggregation design through interaction

In our prototype system, map designers can replace the current set of aggregation patterns by adjusting the design parameters. As mentioned earlier, our design parameters include the threshold for the distance between adjacent buildings and the upper bounds on the three costs. The threshold distance can control the extent of the initial proximity graph over the building features. Lowering the upper bound for the data cost encourages assignment of building features to specific group labels even when the features are spatially distant from the groups, and lowering that for the label cost invites the presence of group labels. Conversely, raising the smoothness cost bound encourages adjacent building features to join the same group label. The candidate aggregation patterns are ordered from top to bottom according to the degree of optimality, as in the vertical histogram, where each row contains the patterns with the total cost in the corresponding bin. We have also equipped our design system with the interface for adjusting the size of this histogram bin to visually improve the arrangement of candidate aggregation patterns. In addition, each aggregation pattern contains the bar graph of the three costs where the data, smoothness, and label bars are colored in red, green, and blue, respectively.

Our system has an interface to force the aggregation of a particular set of manually selected building features when designers cannot find their preferred alternatives (Figure 5). This is accomplished by assigning a small data cost (e.g., 0.1) to the newly specified group label as well as a small data cost (e.g., 0.1) to the selected building features with respect to that group. See the supplementary material for the manual selection of a building group for the complex city block in Figure 1 (Appendix C).



**Figure 6:** Enclosing two rectangular polygons with a concave hull. Samples on the polygon boundaries (in orange) and the two nearest neighbors from the present position (in blue). (a) Moving from the left polygon to the right. (b) Starting to outline the right polygon. (c) Moving back from the right polygon to the left. (d) Keeping to wrap the left polygon. (e) Closing the concave hull.

## 5. Simplifying Boundaries of Aggregated Features

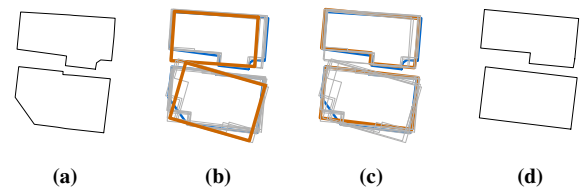
After the aggregation process, we needed to generate a contour profile that encloses an aggregated set of building features. This section describes our strategy for generating appropriate boundary contours for aggregated sets of building features. We also implemented simplification operations on such contour lines as a post-process of building aggregation.

### 5.1. Enclosing features with concave hulls

In our approach, we wanted to cover the aggregated set of building features with a contour that closely fits the boundary of the buildings. This is due to the requirement to eliminate unnecessary conflicts between the building features even after aggregation. Our solution was to use concave hulls that closely follow the concave shapes formed by the aggregated building set. For this purpose, we introduced an algorithm proposed in [MS07].

This method extends the conventional gift-wrapping algorithm for computing convex hulls [Jar73]. Similar to the gift-wrapping algorithm, this algorithm selects the next sample point that forms a straight line from the last sample point in such a way that the sample points are on one (e.g., the left) side of the line. The only difference is that the sample points to be searched are limited to the  $k$ -nearest neighbors of the current sample, so the gift wrapping is done in the local neighborhood.

Figure 6 shows how the algorithm works for the two rectangular polygons, where the samples on the boundary edges are colored in orange. The algorithm starts the wrapping process from the lower left sample and searches for the next sample within a set of  $k$ -nearest neighbors colored in blue, where  $k = 2$  in this case. We switch our step from the left polygon to the right, allowing for the selection of the right blue sample as the next point because it makes the largest counterclockwise polar angle at the current point (Figure 6a). Once we jump to the right polygon, we can continue to follow its boundary (Figure 6b). We do the same local search when we return from the right polygon to the left (Figures 6c and 6d). Finally, we complete the wrapping process for all samples to form the concave hull (Figure 6e). In our implementation, we began computing concave hulls with  $k = 2$  and increment  $k$  if we failed to compose proper hulls. Our experiments showed that increasing the value of  $k$  one by one usually gave us the appropriate concave hulls.



**Figure 7:** Simplifying aggregated building polygons. (a) Aggregated building polygons. (b) Simplified polygon candidates (gray), the original shape (blue), and the current choice (orange). (c) Selecting the simplified shape for each polygon. (d) Finalizing the choices.

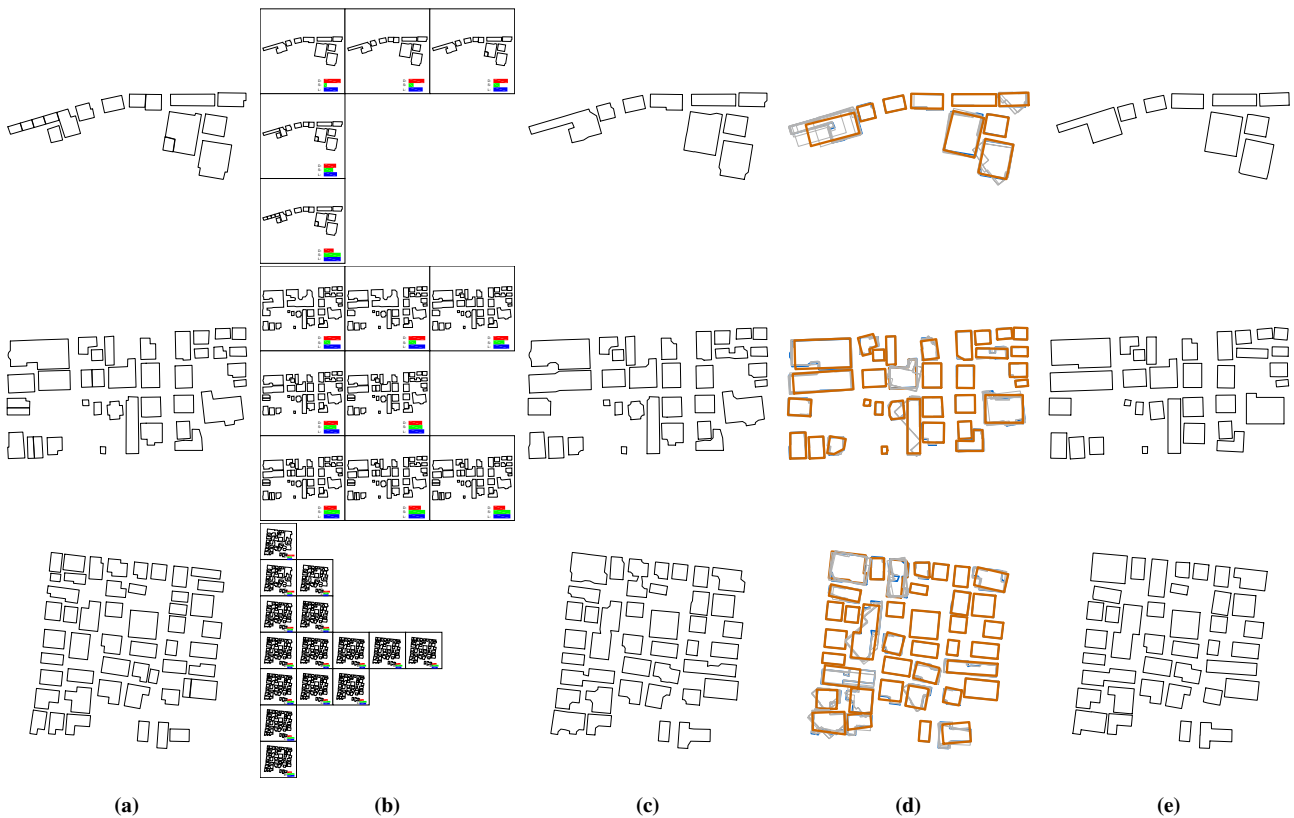
### 5.2. Simplifying the contour of aggregated buildings

As a post-process to the building aggregation, we implemented simplification operations to further clean up the boundary shapes. For this purpose, we used the area-preserving edge-move algorithm formulated by Buchin et al. [BMS11b] (See Appendix D in the supplementary material), in which we made several adjustments to improve the design interactions.

Here, we again collected candidate simplified shapes to help map designers interactively select their preferred shape. This was done by adaptively simplifying the given polygons according to the number of vertices. We also adjusted the choice of principal orientations in squaring the polygon shapes [BMS11b]. This was accomplished by computing the orientation angles of the boundary edges and accumulating their distribution convolved with Gaussians over the cyclic range of angles, which allowed us to extract peaks as representative orientations of the boundary edges. Consider the boundaries of the aggregated building features (Figure 7a). Applying different Gaussian kernels and adaptive polygon simplification produces various results (Figure 7b). Among these candidates, designers can select their own choice for each polygon individually through the interface (Figure 7c). Finally, the designers can fix their choices for simplified shapes (Figure 7d). Each edge-move operation explores the optimal translation of a pair of boundary edges around the building polygon. This usually requires enough free space around the polygon to allow the target edge pair to move sufficiently. However, we limited the displacement tolerance for boundary edges to avoid unnecessary collisions between adjacent building polygons to avoid unnecessary collisions between adjacent building polygons, which sometimes results in insufficiently simplified polygon shapes. We mitigated this problem by automatically shrinking the polygon shape by a certain amount (e.g. 80%) to leave enough margin around it before simplification and then enlarging it after simplification to restore its original size. This polygon resizing for the simplification operations usually works well, but we also provided an interface for cartographers to shrink/enlarge the polygon shapes as needed.

## 6. Results

In this section, we present design examples to demonstrate the feasibility of our approach, followed by evaluation by domain experts and discussion.



**Figure 8:** Cartographic aggregation operations followed by simplification as a post-process. (a) Input features. (b) Candidate aggregation patterns. (c) Selected aggregation. (d) Candidate simplification patterns. (e) Final features after several trial and error simplification designs. Kisarazu area with a sparse layout (top), Ohta area with a semi-dense layout (middle), and Ichikawa area with a dense layout (bottom).

### 6.1. Design examples

We implemented our prototype system on a laptop PC (MacBook Pro) with an Apple M1 Max ten-core CPU, 32GB RAM, and a 32-core GPU. The source code was written in C++ using FLTK for the user interface, OpenGL for map drawing, and CGAL for geometric computation. We also used source code published by DeLong et al. [DOIB10, DOIB12] for collecting locally optimal solutions during label cost optimization. Our system could interactively update a set of aggregation candidates for building features as we adjusted design parameters, due to the fast and inexpensive  $\alpha$ -expansion algorithm. See the accompanying video for actual interactions with our prototype system. The following experimental results are based on map data in Japan, including Digital Road Map Database provided by Sumitomo Electric System Solutions, Co., Ltd. and Residential Map Data ZmapTownII provided by Zenrin Co., Ltd., and obtained through the joint research with the Center for Spatial Information Science of the University of Tokyo (No. 4336).

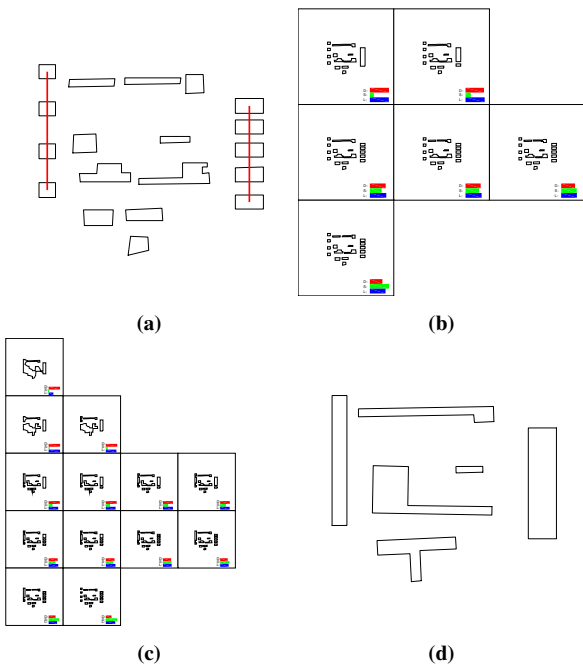
Figure 1 shows how we can interactively update the set of candidate aggregation patterns for building features by adjusting design parameters. Each candidate pattern is highlighted with the hovering of the mouse cursor to clearly display the difference between the original layout of building features and other aggregation pat-

terns. This allows map designers to understand the pros and cons of each design through visual comparison.

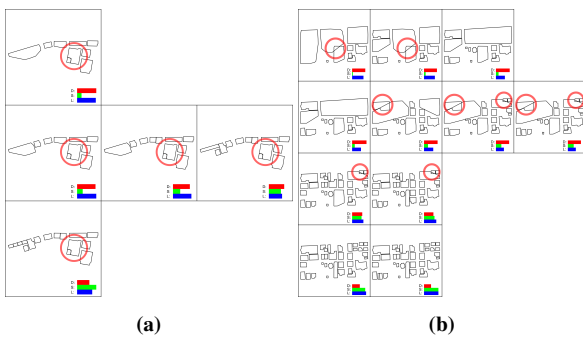
Figure 8 shows several design examples along with the map processing steps. Our prototype system takes building features in the selected city block as input (Figure 8a) and computes aggregation patterns as candidates (Figure 8b). After selecting the optimal candidate (Figure 8c), the system presents simplification options for the aggregated building polygons upon request (Figure 8d). By exploring the best simplification patterns for each building polygon individually, possibly along with further refinement of the building shapes, cartographers can finalize their map design (Figure 8e).

In the first case, with a sparse layout of buildings, we selected a visually plausible aggregation from the initial set of candidates. Note that appropriate simplification options were provided to preserve the shape of the L-shaped aggregation polygons on the left. In the second example, the selected city block has a semi-dense and spatially irregular layout of buildings. In this case, we aggregated only the buildings that are in contact with each other while improving the smoothness of the boundary contours with characteristic shape features left untouched in the simplification process. The last case is a city block with a dense layout of small buildings. In this case, we can adaptively explore the best aggregation choice through interaction with our system. In particular, we manu-





**Figure 9:** Comparison of aggregation results with the previous method [LHDZ13]. (a) Example case shown in [LHDZ13], where the two lines indicate the sets of aggregated buildings. (b) Initial set of aggregation patterns. (c) Updated aggregation patterns by adjusting design parameters. (d) Final results with our approach.



**Figure 10:** Failure cases of the previous approach [NSX\*11] specifically in the context of cartographic aggregation. (a) Two closely adjacent polygons circled in red are never aggregated. (b) Convex polygons incur unwanted conflicts with surroundings as indicated by a red circle. See Figure 8 for a comparison.

ally combined some pairs of building features through the interface to encourage more aggregation of building features. Simplification operations helped us retain the important shape features when we needed to edit landmark buildings.

Figure 9 shows a comparison with the conventional aggregation method based on Gestalt principles for object grouping [LHDZ13]. Here, we reproduced an example layout of building features raised in [LHDZ13] to make the comparison (Figure 9a). The conventional method [LHDZ13] proposed two groups of building fea-

tures to be aggregated as indicated by the red lines, although it did not provide any specific shapes of the contours enclosing the two group. On the other hand, our approach initially proposed a set of aggregation patterns (Figure 9b), which unfortunately did not include the target sequence of buildings as a candidate. However, we obtained the better set of aggregation patterns by interactively adjusting the design parameters (Figure 9c). Finally, we obtained aggregated building features that are consistent with aggregated building with the results obtained by [LHDZ13] (Figure 9d).

Figure 10 exhibits the failure cases generated by the previous approach [NSX\*11], which we used as a basis. Figure 10a reveals that a combination of the two closely adjacent polygons is not proposed as a candidate due to the different definitions of the distance, even when the smaller polygon appears to be part of the larger polygon. Figure 10b offers an example in which the convex outline for aggregating multiple building polygons causes unwanted conflicts with surrounding features. See Figure 8 for a comparison.

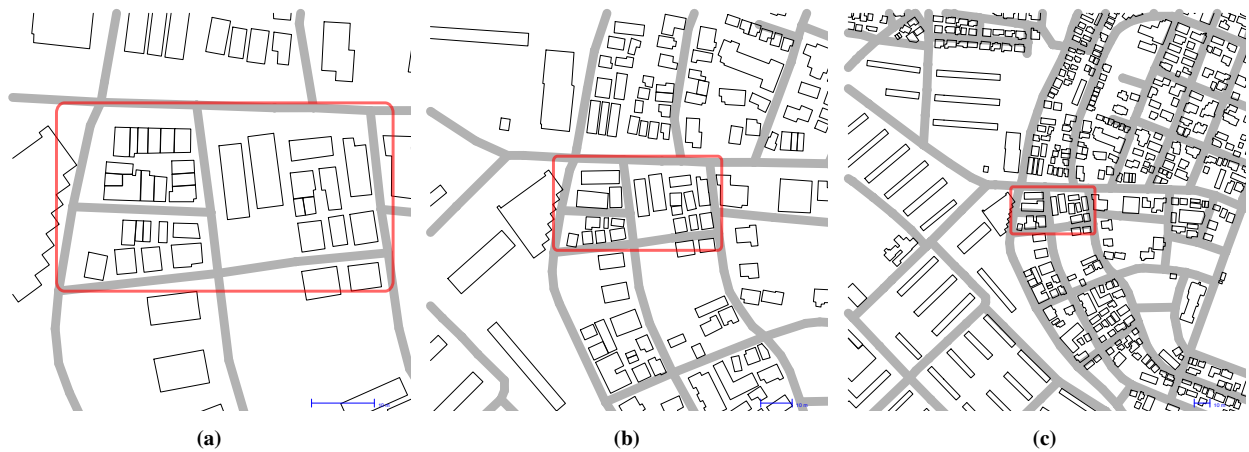
Figure 11 demonstrates that our approach allows the progressive aggregation of the building features as the map scale decreases. In this case, as described in Section 3.2, we initially aggregated building features to some degree at a large map scale and then progressively aggregated them for medium and small map scales.

## 6.2. Feedback from domain experts

We asked two domain experts outside our project team to evaluate the ability of our prototype system to design aggregation patterns on scale-aware maps. During the interview, we first demonstrated our interactive system at the request of the experts and then described the design guidelines (see Section 3.1) as a reference for their evaluation criteria, which is followed by the explanation of the algorithmic details behind the proposed approach. The final interviews were conducted online, but for Expert A, we asked his opinion in person prior to the online interview.

Expert A was a university faculty member specializing in map usability and interfaces of digital maps on mobile devices. He was very knowledgeable about the principles of cartographic generalization, especially from a computational viewpoint. He tested the early version of our design system and understood how it has been improved so far. Expert A appreciated the interface design for explicitly guiding map designers from aggregation to simplification operations. He wanted to compare two or three design candidates in the final selection phase because he wanted to place these design choices on the map to verify that they fit the surrounding map configuration. He also wanted a design operation that explicitly prohibited the aggregation of a particular set of features. In practice, he advised us to preserve important features of building shapes when applying simplification operations in the early stage of this study. Thus, we decided to include polygonal shapes with a small number of corners (see Section 3.1) in the set of candidate simplification patterns in our system.

Expert B was familiar with the mechanisms of cartographic generalization because he compiled actual map designs in the Geospatial Information Authority. He was also a researcher in computer science and engineering. He suggested that the proposed aggregation and simplification design schemes have a significant impact on



**Figure 11:** Designing progressive aggregation according to the map scale. (a) Large scale. (b) Medium scale. (c) Small scale. Note that the building features enclosed in red are adaptively aggregated as the map scale reduces. Kashiwa area.

the production of small scale maps from large scale maps. He noted out that the system is complete, and the approach seems promising, so many map designers will benefit from the results.

Regarding the algorithm, Expert B pointed out the importance of keeping the right angles of the buildings when simplifying the boundary profiles. He requested that we improve such operations since our system could not fully refine the orientations of the simplified boundary edges especially when the building boundary has complex geometry. He wanted these simplified edges to be adjusted to match the orientation of the roads outside the city block or to follow the dominant directions of the block derived from the surrounding roads. He specifically wanted to improve the shape quality of landmark buildings such as schools and hospitals through the simplification process. He agreed that the ability to semi-automatically propose a large number of aggregation candidates was highly effective in reducing the workload of map designers, as proposing a unique solution only often does not suffice individual requirements and preferences of such designers.

### 6.3. Discussion

Our proposed approach is suitable for the progressive grouping of adjacent building features because it allows flexible control over the proximity between geographic features. Thus, our approach can facilitate the hierarchical aggregation of geographic features and presents acceptable solutions for scale-aware maps (Figure 11).

The design parameters help us explore different candidate patterns in the aggregation of building features. Although it is possible to provide general hints for adjusting such parameters by referring to the formulation of label cost optimization, developing more specific guidelines for automatically adjusting parameters according to the target geographic features and required design rules remains a technical challenge. A possible solution is to collect the actual selection of design parameters to track the relationship with the configuration of geographic features and design principles.

Preserving the shape characteristics of landmark buildings has a

significant impact on the map design scenario because we need to respect their original shapes even during aggregation and simplification operations. Our approach does not fully cover such problems when the input set of buildings contains specific landmarks such as schools, hospitals, and supermarkets. Improving the formulation for cartographic aggregation by taking into account the semantics of building types is left to future work.

Although map designers follow general rules for balancing the physical shape and visual appearance of a map, they have specific preferences and design philosophies for designing shape details. The rules also slightly differ from country to country since the density and, thus, the layout patterns of building features are different in each country. Our approach to presenting a set of candidates is effective in accommodating such differences among designers and countries, but presenting candidates that better meet specific needs is an important topic for further research.

### 7. Conclusion

This paper has presented an approach for aggregating building features in the context of scale-aware map design. Rather than proposing a unique solution, our approach suggests a variety of candidate aggregation patterns to assist a wide range of cartographers with unique design preferences through interactive design. This is achieved by collecting locally optimal solutions in the course of label cost optimization. By adjusting design parameters, we can interactively explore the better sets of design choices. Introducing concave hulls to wrap the aggregated set of building features not only prevents unwanted design artifacts but also refines the aggregation results. Simplification to improve the shape quality of building contours has also been implemented as a post-process. The capability of the proposed approach was demonstrated with design examples, evaluation by two domain experts, and discussion of its limitations.

### Acknowledgments

This work is supported by JSPS KAKENHI Grant Numbers 19H04120 and 24K02981.

## References

- [AS01] AGRAWALA M., STOLTE C.: Rendering effective route maps: Improving usability through generalization. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 241–250. doi:10.1145/383259.383286. 3
- [BBW05] BADER M., BARRAULT M., WEIBEL R.: Building displacement over a ductile truss. *International Journal of Geographical Information Science* 19 (2005), 915–936. doi:10.1080/13658810500161237. 3
- [BC07] BURGHARDT D., CECCONI A.: Mesh simplification for building typification. *International Journal of Geographical Information Science* 21, 3 (2007), 283–298. doi:10.1080/13658810600912323. 3
- [BMS11a] BUCHIN K., MEULEMANS W., SPECKMANN B.: Area-preserving c-oriented schematization. In *Abstracts of the 27th European Workshop on Computational Geometry* (2011), pp. 163–166. 3
- [BMS11b] BUCHIN K., MEULEMANS W., SPECKMANN B.: A new method for subdivision simplification with applications to urban-area generalization. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2011), pp. 261–270. doi:10.1145/2093973.2094009. 3, 7
- [BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (2001), 1222–1239. doi:10.1109/34.969114. 5
- [CvDH14] CHIMANI M., VAN DIJK T. C., HAUNERT J.-H.: How to eat a graph: Computing selection sequences for the continuous generalization of road networks. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2014), pp. 243–252. doi:10.1145/2666310.2666414. 3
- [DOIB10] DELONG A., OSOKIN A., ISACK H. N., BOYKOV Y.: Fast approximate energy minimization with label costs. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), pp. 2173–2180. doi:10.1109/CVPR.2010.5539897. 5, 6, 8
- [DOIB12] DELONG A., OSOKIN A., ISACK H. N., BOYKOV Y.: Fast approximate energy minimization with label costs. *International Journal of Computer Vision* (2012), 1–27. doi:10.1007/s11263-011-0437-z. 5, 6, 8
- [FTS19] FENG Y., THIEMANN F., SESTER M.: Learning cartographic building generalization with deep convolutional neural networks. *ISPRS International Journal of Geo-Information* 8 (2019). doi:10.3390/ijgi8060258. 3
- [GASP08] GRABLER F., AGRAWALA M., SUMNER R. W., PAULY M.: Automatic generation of tourist maps. *ACM Transactions on Graphics* 27, 3 (2008). doi:10.1145/1360612.1360699. 3
- [GW18] GONG X., WU F.: A typification method for linear pattern in urban building generalisation. *Geocarto International* 33, 2 (2018), 189–207. doi:10.1080/10106049.2016.1240718. 3
- [HS11] HAUNERT J.-H., SERING L.: Drawing road networks with focus regions. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2555–2562. doi:10.1109/TVCG.2011.191. 3
- [HWAT13] HIRONO D., WU H.-Y., ARIKAWA M., TAKAHASHI S.: Constrained optimization for disoccluding geographic landmarks in 3D urban maps. In *Proceedings of the 6th IEEE Pacific Visualization Symposium (PacificVis 2013)* (2013), pp. 17–24. doi:10.1109/PacificVis.2013.6596123. 3
- [Jar73] JARVIS R. A.: On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters* 2, 1 (1973), 18–21. doi:10.1016/0020-0190(73)90020-3. 7
- [KAB\*10] KOPF J., AGRAWALA M., BARGERON D., SALESIN D., COHEN M.: Automatic generation of destination maps. *ACM Transactions on Graphics* 29, 6 (2010). doi:10.1145/1882261.1866184. 3
- [KDE05] KULIK L., DUCKHAM M., EGENHOFER M.: Ontology-driven map generalization. *Journal of Visual Languages and Computing* 16, 3 (2005), 245–267. doi:10.1016/j.jvlc.2005.02.001. 3
- [Lee96] LEE D.: *Automation of Map Generalization: The Cutting-Edge Technology*. Tech. rep., Environmental Systems Research Institute, Inc. (ESRI), 1996. 2
- [LGSM14] LIU Y., GUO Q., SUN Y., MA X.: A combined approach to cartographic displacement for buildings based on skeleton and improved elastic beam algorithm. *PLOS ONE* 9 (12 2014), 1–26. doi:10.1371/journal.pone.0113953. 3
- [LHDZ13] LIQIANG Z., HAO D., DONG C., ZHEN W.: A spatial cognition-based urban building clustering approach and its applications. *International Journal of Geographical Information Science* 27, 4 (2013), 721–740. doi:10.1080/13658816.2012.700518. 2, 9
- [Li06] LI Z.: *Algorithmic Foundation of Multi-Scale Spatial Representation*. CRC Press, 2006. 2
- [LJ01] LONERGAN M., JONES C.: An iterative displacement method for conflict resolution in map generalization. *Algorithmica* 30 (2001), 287–301. doi:10.1007/s00453-001-0011-0. 3
- [LYAC04] LI Z., YAN H., AI T., CHEN J.: Automated building generalization based on urban morphology and gestalt theory. *International Journal of Geographical Information Science* 18, 5 (2004), 513–534. doi:10.1080/13658810410001702021. 2
- [May05] MAYER H.: Scale-spaces for generalization of 3d buildings. *International Journal of Geographical Information Science* 19, 8-9 (2005), 975–997. doi:10.1080/13658810500161286. 3
- [MS92] MCMASTER R. B., SHEA K. S.: *Generalization in Digital Cartography*. Association of American Geographers, 1992. 2
- [MS07] MOREIRA A., SANTOS M.: Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. In *Proceedings of the Second International Conference on Computer Graphics Theory and Applications (GRAPP2007)* (2007), pp. 61–68. 2, 7
- [MTW\*19] MARUYAMA K., TAKAHASHI S., WU H.-Y., MISUE K., ARIKAWA M.: Scale-aware cartographic displacement based on constrained optimization. In *Proc. of 23rd International Conference Information Visualisation (IV2019)* (2019), pp. 74–80. doi:10.1109/IV.2019.00022. 3
- [NSX\*11] NAN L., SHARF A., XIE K., WONG T.-T., DEUSSEN O., COHEN-OR D., CHEN B.: Conjoining gestalt rules for abstraction of architectural drawings. *ACM Transactions on Graphics* 30, 6 (2011). doi:10.1145/2070781.2024219. 2, 3, 4, 5, 9
- [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. doi:10.1109/TVCG.2010.81. 3
- [PK19] PILEHFOROOSHHA P., KARIMI M.: An integrated framework for linear pattern extraction in the building group generalization process. *Geocarto International* 34, 9 (2019), 1000–1021. doi:10.1080/10106049.2018.1458253. 2
- [PY11] PARK W., YU K.: Hybrid line simplification for cartographic generalization. *Pattern Recognition Letters* 32, 9 (2011), 1267–1273. doi:10.1016/j.patrec.2011.03.013. 3
- [QWC\*09] QU H., WANG H., CUI W., WU Y., CHAN M.-Y.: Focus+Context route zooming and information overlay in 3D urban environments. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1547–1554. doi:10.1109/TVCG.2009.144. 3
- [Reg01] REGNAULD N.: Contextual building typification in automated map generalization. *Algorithmica* 30 (2001), 312–333. doi:10.1007/s00453-001-0008-8. 3
- [RF99] RADKE J., FLODMARK A.: The use of spatial decompositions for constructing street centerlines. *Geographic Information Sciences* 5, 1 (1999), 15–23. doi:10.1080/10824009909480509. 4

- [RMM\*06] ROBINSON A. H., MORRISON J. L., MUEHRCKE P. C., KIMERLING A. J., GUPTILL S. C.: *Elements of Cartography*, 6th ed. John Wiley & Sons, Inc., 2006. 2
- [Rua98] RUAS A.: A method for building displacement in automated map generalisation. *International Journal of Geographical Information Science* 12, 8 (1998), 789–803. doi:10.1080/136588198241509.3
- [Sch08] SCHMID F.: Knowledge-based wayfinding maps for small display cartography. *Journal of Location Based Services* 2, 1 (2008), 57–83. doi:10.1080/17489720802279544.3
- [Ses05] SESTER M.: Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science* 19 (2005), 871–897. doi:10.1080/13658810500161179.3
- [SLW\*22] SHEN Y., LI J., WANG Z., ZHAO R., WANG L.: A raster-based typification method for multiscale visualization of building features considering distribution patterns. *International Journal of Digital Earth* 15, 1 (2022), 249–275. doi:10.1080/17538947.2021.2023668.3
- [SM89] SHEA K. S., MCMASTER R. B.: Cartographic generalization in a digital environment: When and how to generalize. In *Proceedings Auto-Carto 9* (1989), pp. 56–67. 2
- [SvKF06] SPECKMANN B., VAN KREVELD M., FLORISSON S.: A linear programming approach to rectangular cartograms. In *Progress in Spatial Data Handling: 12th International Symposium on Spatial Data Handling* (2006), pp. 529–546. doi:10.1007/3-540-35589-8\_34.3
- [vDvGH\*13] VAN DIJK T., VAN GOETHEM A., HAUNERT J.-H., MEULEMANS W., SPECKMANN B.: Accentuating focus maps via partial schematization. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2013), pp. 428–431. doi:10.1145/2525314.2525452.3
- [vGMR\*13] VAN GOETHEM A., MEULEMANS W., REIMER A., HAVERKORT H., SPECKMANN B.: Topologically safe curved schematization. *The Cartographic Journal* 50 (2013), 276–285. doi:10.1179/1743277413Y.0000000066.3
- [vGMSW14] VAN GOETHEM A., MEULEMANS W., SPECKMANN B., WOOD J.: Exploring curved schematization. In *Proceedings of the 7th IEEE Pacific Visualization Symposium (PacificVis 2014)* (2014), pp. 1–8. doi:10.1109/PacificVis.2014.11.3
- [vGMSW15] VAN GOETHEM A., MEULEMANS W., SPECKMANN B., WOOD J.: Exploring curved schematization of territorial outlines. *IEEE Transactions on Visualization and Computer Graphics* 21, 8 (2015), 889–902. doi:10.1109/TVCG.2015.2401025.3
- [WC11] WANG Y.-S., CHI M.-T.: Focus+context metro maps. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2528–2535. doi:10.1109/TVCG.2011.205.3
- [WGL\*17] WANG L., GUO Q., LIU Y., SUN Y., WEI Z.: Contextual building selection based on a genetic algorithm in map generalization. *ISPRS International Journal of Geo-Information* 6, 9 (2017). doi:10.3390/ijgi6090271.3
- [WJT03] WARE J. M., JONES C. B., THOMAS N.: Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science* 17, 8 (2003), 743–769. doi:10.1080/13658810310001596085.3
- [WTH\*13] WU H.-Y., TAKAHASHI S., HIRONO D., ARIKAWA M., LIN C.-C., YEN H.-C.: Spatially efficient design of annotated metro maps. *Computer Graphics Forum* 32, 3 (2013), 261–270. doi:10.1111/cgf.12113.3
- [WTLY12] WU H.-Y., TAKAHASHI S., LIN C.-C., YEN H.-C.: Travel-route-centered metro map layout and annotation. *Computer Graphics Forum* 31, 3 (2012), 925–934. doi:10.1111/j.1467-8659.2011.02053.x.3
- [WWW03] WILSON I. D., WARE J. M., WARE J. A.: A genetic algorithm approach to cartographic map generalisation. *Computers in Industry* 52, 3 (2003), 291–304. doi:10.1016/S0166-3615(03)00132-5.3
- [XAY\*23] XIAO T., AI T., YU H., YANG M., LIU P.: A point selection method in map generalization using graph convolutional network model. *Cartography and Geographic Information Science* (2023), 1–21. doi:10.1080/15230406.2023.2187886.3
- [YAY\*22] YAN X., AI T., YANG M., TONG X., LIU Q.: A graph deep learning approach for urban building grouping. *Geocarto International* 37, 10 (2022), 2944–2966. doi:10.1080/10106049.2020.1856195.3
- [YYY\*22] YANG M., YUAN T., YAN X., AI T., JIANG C.: A hybrid approach to building simplification with an evaluator from a backpropagation neural network. *International Journal of Geographical Information Science* 36, 2 (2022), 280–309. doi:10.1080/13658816.2021.1873998.3