International Journal of Digital Earth

ISSN: 1753-8947 (Print) 1753-8955 (Online) Journal homepage: www.tandfonline.com/journals/tjde20

# Point label placement in 3D geographic scene via elastic beam algorithm

Zhiwei Wei, Nai Yang, Wenjia Xu, Su Ding, Yebin Chen & Renzhong Guo

**To cite this article:** Zhiwei Wei, Nai Yang, Wenjia Xu, Su Ding, Yebin Chen & Renzhong Guo (2025) Point label placement in 3D geographic scene via elastic beam algorithm, International Journal of Digital Earth, 18:2, 2563208, DOI: 10.1080/17538947.2025.2563208

**To link to this article:** https://doi.org/10.1080/17538947.2025.2563208

INTERNATIONAL JOURNAL OF DIGITAL EARTH
2025, VOL. 18, NO. 2, 2563208
https://doi.org/10.1080/17538947.2025.2563208

# Point label placement in 3D geographic scene via elastic beam algorithm

Zhiwei Wei[a,b,c], Nai Yang[d], Wenjia Xu[e], Su Ding[f], Yebin Chen[g] and Renzhong Guo[c,g]

[a]Hunan Key Laboratory of Geospatial Big Data Mining and Application, Hunan Normal University, Changsha, People's Republic of China; [b]School of Geographical Sciences, Hunan Normal University, Changsha, People's Republic of China; [c]Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen, People's Republic of China; [d]School of Geography and Information Engineering, China University of Geosciences (Wuhan), Wuhan, People's Republic of China; [e]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, People's Republic of China; [f]College of Environmental and Resource Sciences, College of Carbon Neutral Zhejiang A&F University, Hangzhou, People's Republic of China; [g]Research Institute for Smart City, School of Architecture and Urban Planning, Shenzhen University, Shenzhen, People's Republic of China

**ABSTRACT**
Labels are essential for conveying spatial information in maps, and their effective placement is critical for readability and clarity. In 3D geographical scenes, point labels are commonly displayed with leader lines to enhance visibility, providing flexibility in their positioning. Although the labels are rendered in 3D space, their placement optimization is often carried out in the 2D projected screen space, based on their screen coordinates. However, existing methods primarily rely on heuristic approaches or candidate position models to place the labels, leading to inefficient map space utilization and neglecting the relative relationships between labels. To address this issue, we propose a novel approach that conceptualizes leadered label positioning as a label displacement problem and applies the elastic beam displacement algorithm from cartographic generalization to solve it. In this approach, a triangulated graph is built as the elastic beam structure to delineate the spatial relationships among labels. Labels that violate label quality requirements are treated as applying forces on the beams' nodes. Then the elastic beam algorithm is iteratively applied to assign forces to each node, determining the new positions of labels. Our experimental results demonstrate that this method effectively avoids label overlays while maintaining minimal directional deviation between adjacent labels.

**ARTICLE HISTORY**
Received 27 February 2025
Accepted 24 August 2025

**KEYWORDS**
3D Visualization, label placement, map generalization, displacement algorithm, minimum energy algorithm

## 1. Introduction

In modern mapping applications, such as 3D terrain maps (Chen et al. 2010), or augmented reality maps (Gemsa, Haunert, and Nöllenburg 2015; Tatzgern et al. 2014), placing labels related to the landmarks clearly is essential to help users understand spatial information. For example, in a 3D map of a mountainous region, labels for peaks or scenic spots must be placed clearly without obstructing the terrain or overlapping each other, as shown in Figure 1(a). Similarly, in a 3D city scene, labels for iconic buildings such as towers or shopping centers need to be positioned in a way that avoids clutter while remaining visually associated with the correct locations, as shown in Figure 1(b). As a result, a variety of methods have been developed to automate map labeling. These methods can be classified based on the type of spatial features they address – points, lines, and areas (Li et al. 2018; Wei et al. 2025). Among these, point feature labeling presents the greatest challenge due to its complexity, making it a primary focus of research (Fan et al. 2023; Wolff and Strijk 1999).

The core objective of point label placement is to generate an overlap-free and visual-coherent label layout across various applications (Dijk et al. 2002; Fan, Zhang, and Du 2005). To ensure visual coherence, fixed or slider position models are commonly applied to keep labels close to their associated points (Christensen, Marks, and Shieber 1995; Li et al. 2018; van Kreveld, Strijk, and Wolff 1998). Building upon these position models, various heuristic and optimization methods, such as simulated annealing, tabu search, genetic algorithms, ant colony optimization and reinforcement learning, have been applied to produce overlap-free layouts (Bobák, Čmolík, and Čadík 2023; Gomes, Lorena, and Ribeiro 2016; Lu et al. 2019; Peng, Song, and Wu

**CONTACT** Zhiwei Wei 2011301130108@whu.edu.cn No. 36, Lushan South Road, Yuelu District, Changsha City, Hunan Province, People's Republic of China

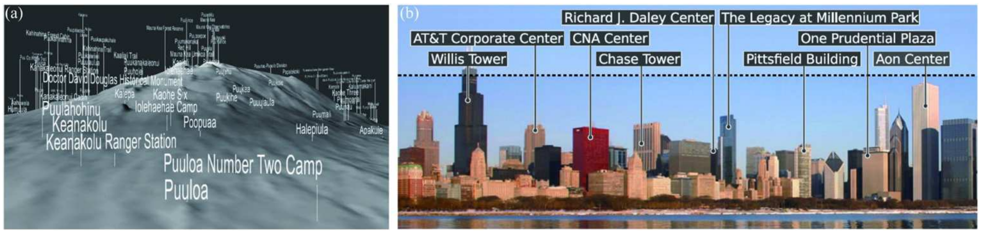


**Figure 1.** Examples of labels with leaders in 3D geographic scene. (a) Labels for the Island of Hawaii (Chen et al. 2010); (b) Skyline labels for Chicago City (Gemsa, Haunert, and Nöllenburg 2015).

2007; Zheng, Guo, and Liu 2006; Zhou et al. 2015; Zoraster 1997). To enhance computational efficiency, some studies have also introduced techniques like clustering to divide elements into subgroups or organizing label placement based on label density (Araujo, Chaves, and Lorena 2019; Cao et al. 2022; Lessani et al. 2025; Lessani, Deng, and Guo 2021; Zhou et al. 2016). However, these methods predominantly address two-dimensional (2D) maps, with far less attention devoted to three-dimensional (3D) scene labeling. As 3D applications continue to gain popularity, the need for effective 3D label placement is becoming increasingly critical (Dong et al. 2020; She et al. 2019).

In contrast to 2D maps, where labels are typically placed near features, 3D scenes often rely on leader lines to connect labels to their associated features (Bobák, Čmolík, and Čadík 2020; Chen et al. 2010; Gemsa, Haunert, and Nöllenburg 2015; Su et al. 2023). This approach helps mitigate issues such as occlusion and enhances the referential clarity between labels and their corresponding features (as shown in Figure 1). The added flexibility in 3D label placement introduces new challenges, rendering traditional fixed or slider position models unsuitable for 3D environments. Consequently, heuristic methods and force-directed approaches have been employed to adjust label positions and avoid overlap (Chen et al. 2010; Hao and Lin 2022; Maass and Döllner 2006; Wei et al. 2025). However, these methods tend to displace labels individually, which can significantly distort the relative spatial relationships between labels. This highlights the need for new approaches that can balance all quality requirements in 3D label placement.

Displacement is a fundamental operation in cartographic generalization, and several algorithms have been developed to address this task. Among them, the elastic beam algorithm is particularly powerful, as it enables global optimization of map quality requirements through energy minimization (Bader 2001; Liu et al. 2014; Wei et al. 2023). If we treat the connections between labels as elastic beams, and the requirements for 3D label placement as map quality requirements. Then we can also apply the elastic beam algorithm to calculate the ideal label placement. As the algorithm allows for the global optimization of label positions while adhering to quality constraints such as overlap avoidance and maintenance of relative relationships, it can provide a powerful solution for 3D label placement.

Motivated by the above thoughts, we applied the elastic beam algorithm to 3D label placement. In this approach, labels are connected through elastic beams to represent their proximity relationships. Attractive and repulsive forces are calculated for each beam to dynamically adjust the labels' positions, ensuring that overlaps are avoided and spatial relationships are preserved. These forces are mathematical representations of the label placement requirements. Labels that violate these requirements are subjected to forces that guide their repositioning. By iteratively adjusting the label positions according to these forces, the elastic beam algorithm can achieve optimal 3D label placement, improving clarity and user experience in complex 3D visualizations. Note that although our method is applied to 3D geographic scenes, all placement computations are performed in the 2D screen space. This is consistent with existing displacement models, such as the elastic beam algorithm, which operates in projected 2D coordinates.

## 2. Related works

### 2.1. Quality requirements for point label placement in 3D geographic scene

When placing labels in 3D geographic scenes, it is essential to consider the unique characteristics of 3D visualization while adhering to cartographic principles (She et al. 2019). This paper summarizes the requirements for point label placement with leader lines in 3D geographic scenes. In this context, the leader lines refer to those with a fixed direction and an unfixed anchor point for the label, as this configuration is widely used in practical applications (Wei et al. 2025). However, other types of leader lines may also be employed in practices, and our approach can be extended to these variations. We will further elaborate on this in *Section* 5.1.

#### 2.1.1. Geometry

Position ($\boldsymbol{GR_1}$): The label's position should not deviate excessively laterally from its corresponding point feature, as this would disrupt the connection between the label and its leader line (Tatzgern et al. 2014).

Size ($\boldsymbol{GR_2}$): The label's size must meet the perspective requirements of 3D scenes, where nearer objects appear larger and distant objects smaller (Zhou et al. 2016). Additionally, the label should be large enough to be legible, such as a minimum of 2pt on screen (He, Song, and Li 2016).

#### 2.1.2. Relation

Overlap avoidance ($\boldsymbol{RR_1}$): Labels should not overlap with other labels or existing map elements (e.g. point features) (Christensen, Marks, and Shieber 1995).

Minimum distance maintenance ($\boldsymbol{RR_2}$): the distance between labels or between labels and existing map elements (e.g. point features) should exceed a minimum threshold, $d_{\min}$. For instance, at a view distance of 30 cm, the human eye can discern a minimum separation of 0.2 mm (Liu et al. 2014).

Relative relation maintenance ($\boldsymbol{RR_3}$): The spatial directional relationships between labels should, as much as possible, reflect the spatial directional relationships between the corresponding point features (Fan et al. 2023).

#### 2.1.3. Distribution

Total displacement distance ($\boldsymbol{DR_1}$): The total distance that labels are displaced should be minimized as much as possible (Bobák, Čmolík, and Čadík 2020).

These requirements may sometimes conflict with or reinforce one another. For instance, displacing labels to avoid overlap ($\boldsymbol{RR_1}$) may increase the total displacement distance ($\boldsymbol{DR_1}$). Therefore, the label placement requirements should be expressed in an optimized form, meaning they should be fulfilled as much as possible. Furthermore, the requirements may carry different priorities. For example, $\boldsymbol{RR_1}$ and $\boldsymbol{RR_2}$ may take precedence over $\boldsymbol{RR_3}$ because avoiding overlap ($\boldsymbol{RR_1}$) and maintaining a minimum distance ($\boldsymbol{RR_2}$) are essential for preventing visual clutter, while maintaining relative spatial relationships ($\boldsymbol{RR_3}$) is desirable but less critical. It is also important to note that other requirements may also be necessary in practical applications, such as taking into account the structural characteristics of the scene (She et al. 2019).

### 2.2. Algorithms for point label placement

An algorithm for point label placement involves two key components: the position model and the placement strategy. The position model determines the available placement locations for a label, while the placement strategy selects the optimal position from the available options.

#### 2.2.1. Position models

Point label position models can be broadly classified into three types: fixed candidate position models, sliding position models, and random position models. Fixed candidate position models use predefined locations around each point, typically offering options like 4-candidate or 8-candidate positions (Christensen, Marks, and Shieber 1995; Fan et al. 2023; Wei et al. 2025). This model is simple and computationally efficient, but it may not fully optimize the available map space. Sliding position models allow for dynamic adjustment of label placements by sliding them along fixed orientations around the point (Li et al. 2018; van Kreveld, Strijk, and Wolff 1998). This model provides greater flexibility in accommodating varying map features and label densities. However, as the number of map elements increases, the computational complexity of this approach also grows. Compared to random position models, which offer the greatest flexibility, both fixed candidate and sliding position models remain simpler in structure, making them widely used for 2D maps (Wei et al. 2025). Random position models do not impose strict rules on label locations, allowing labels to be placed freely around their corresponding points. To maintain clear associations, labels are often connected to points with leader lines (Bobák, Čmolík, and Čadík 2020; Chen et al. 2010; Gemsa, Haunert, and Nöllenburg 2015). Since labels in this model are not restricted to positions close to the point, random position models are particularly useful in mitigating issues such as occlusion, especially in 3D scenes or dynamic visualizations. As a result, this model or a hybrid of models is frequently employed in 3D environments to strike a balance between flexibility and clarity (She et al. 2019).

In summary, fixed candidate and sliding position models offer simplicity but limit labels to positions close to their corresponding points, making them ideal for 2D maps. On the other hand, random position models provide greater flexibility by allowing labels to be positioned more freely, which is advantageous in 3D environments or dynamic scenarios. In this work, we adopt the random position model for 3D geographic scenes, with labels connected to their respective points using leader lines.

#### 2.2.2. Placement strategies

Earlier point label placement strategies were primarily developed for 2D maps using fixed candidate or sliding position models, and can be broadly categorized into two types: local and global view algorithms. Local view algorithms place labels sequentially, one by one, using heuristic methods such as simulated annealing and tabu search algorithms (Zheng, Guo, and Liu 2006; Zoraster 1997). These strategies offer simplicity and efficiency but are prone to issues like getting stuck in local optima, which may lead to unsatisfactory label layouts. Global view algorithms, on the other hand, attempt to optimize label placement across the entire map by resolving all constraints–such as overlap avoidance–using advanced techniques like genetic algorithms, ant colony optimization, or reinforced learning (Bobák, Čmolík, and Čadík 2023; Gomes, Lorena, and Ribeiro 2016; Lu et al. 2019; Peng, Song, and Wu 2007). While these methods often yield more optimized results, their computational complexity grows significantly with the number of elements, leading to reduced efficiency. To mitigate this, some researchers have explored techniques such as clustering to divide elements into subgroups or organizing label placement based on label density to enhance efficiency (Araujo, Chaves, and Lorena 2019; Cao et al. 2022; Lessani et al. 2025; Lessani, Deng, and Guo 2021; Zhou et al. 2016).

With the growing popularity of 3D applications, researchers have begun adapting these 2D strategies for use in 3D environments. For example, Zhou et al. (2016) applied genetic algorithms to the placement of point feature labels in 3D scenes, accounting for the perspective effect where closer objects appear larger and distant objects appear smaller. She et al. (2019) introduced heuristic methods that consider the semantic structure of 3D scenes for placing building labels. However, these methods rely on fixed candidate models, requiring labels to remain close to their corresponding point features, which can lead to referential ambiguity, especially in 3D scenes with frequent viewpoint changes. To address this limitation, some scholars have adopted the random position model with leader lines to connect point features to their labels, improving the consistency of the label-to-feature relationship. For instance, Chen et al. (2010) proposed a scale-adaptive label placement method with leader lines for point features, effectively labeling geographic names in 3D terrain scenes. Additionally, Hao and Lin (2022) compared leader line-based labels with texture-attached labels in 3D indoor environments, demonstrating that leader line-based labels more clearly and comprehensively convey target information. However, the majority of these leader line-based labeling methods remain heuristic, meaning they do not fully optimize the use of available space or account for the relative relationships between labels. Although Bobák, Čmolík, and Čadík (2020) proposed a dynamic programming strategy for placing building labels with leader lines, this method restricts labels to predefined rows, limiting flexibility. However, it is important to note that most 3D scenes are ultimately presented to users through a 2D screen. Although labels are rendered in 3D space, the above placement optimization strategies are typically conducted in the 2D projected screen space.

In summary, while traditional 2D labeling strategies-both local and global-have been adapted to 3D environments, they face challenges when applied to complex 3D scenes due to their applied fixed candidate position models. The introduction of a random position model with leader lines has improved the consistency of label-to-feature associations in 3D, but most existing approaches are still heuristic-based, which limits their effectiveness. To address these challenges, there is a need to develop new methods for placing point feature labels with leader lines in 3D geographic scenes that improve placement quality by making better use of available space and preserving relative label relationships.

### 2.3. The elastic beam algorithm

The elastic beam algorithm, originally proposed by Bader (2001) and later refined by Liu et al. (2014), provides a physically inspired method to optimize spatial layouts through energy minimization. To help understand its working principle, imagine the relationships between buildings (or labels) as beams or springs connecting them. As shown in Figure 2(a), a single beam responds to external forces by stretching or bending. Accordingly, when these buildings get too close, or too far from where they should be, the connecting beams are also bent or stretched – just like elastic rods responding to external forces. In this algorithm, the spatial proximity between buildings is modeled as a graph, where each edge represents an elastic beam, as shown in Figure 2(b). When placement constraints are violated, forces are applied to the endpoints of the beams, causing the beams to stretch or bend and driving positional adjustments, as shown in Figure 2(c). The forces and moments at each beam's two end nodes are represented as ($f_1{}^x$, $f_1{}^y$, $M_1$) and ($f_2{}^x$, $f_2{}^y$, $M_2$). These forces result in the beams stretching and bending, with the displaced distances and rotations at the two end nodes denoted by ($d_1^x$, $d_1^y$, $\theta_1$) and ($d_2^x$, $d_2^y$, $\theta_2$). The moments ($M_1$ and $M_2$) on the two end nodes are defined as Eq. (1), where ($x_1$, $y_1$), and ($x_2$, $y_2$) are the coordinates of two end nodes (Liu et al. 2014).

$$\begin{cases} M_1 = (y_1 - y_2)*f1^x + (x_1 - x_2)*f_1^y \\ M_2 = (y_2 - y_1)*f_2^x + (x_2 - x_1)*f_2^y \end{cases} \tag{1}$$

The deformation of the beams is expressed in terms of energy, and the algorithm determines the optimal beam shape and position by minimizing this energy, which in turn yields the new locations of the displaced buildings. To solve the energy minimization problem, Finite Element Analysis is applied and the resulting displacements for each building ($d$) can finally be formulated as a matrix, as shown in Eq. (2). Note: the displacements are all calculated only in $x$ and $y$ directions, consistent with 2D screen coordinates, where the elastic beam algorithm is applied. For more details about the mathematical derivation process see Bader (2001) and Liu et al. (2014).

$$\begin{bmatrix} d_{x1} \\ d_{y1} \\ \theta_1 \\ d_{x2} \\ d_{y2} \\ \theta_2 \end{bmatrix} = \frac{l}{E} \begin{bmatrix} Ac^2 + \frac{12I}{l^2}s^2 & (A - \frac{12I}{l^2})cs & -\frac{6I}{l}s & -Ac^2 - \frac{12I}{l^2}s^2 & (A - \frac{12I}{l^2})cs & -\frac{6I}{l}s \\ (A - \frac{12I}{l^2})cs & As^2 + \frac{12I}{l^2}c^2 & \frac{6I}{l}c & -(A - \frac{12I}{l^2})cs & -As^2 - \frac{12I}{l^2}c^2 & \frac{6I}{l}c \\ -\frac{6I}{l}s & \frac{6I}{l}c & 4I & \frac{6I}{l}s & -\frac{6I}{l}c & 2I \\ -Ac^2 - \frac{12I}{l^2}s^2 & -(A - \frac{12I}{l^2})cs & \frac{6I}{l}s & Ac^2 + \frac{12I}{l^2}s^2 & (A - \frac{12I}{l^2})cs & \frac{6I}{l}s \\ -Ac^2 - \frac{12I}{l^2}s^2 & -As^2 + \frac{12I}{l^2}c^2 & -\frac{6I}{l}c & (A - \frac{12I}{l^2})cs & As^2 + \frac{12I}{l^2}c^2 & -\frac{6I}{l}c \\ \frac{-6I}{l}s & \frac{6I}{l}c & 2I & -\frac{6I}{l}s & -\frac{6I}{l}c & 4I \end{bmatrix}^{-1} \begin{bmatrix} f_1^x \\ f_2^y \\ M_1 \\ f_2^x \\ f_2^y \\ M_2 \end{bmatrix} \tag{2}$$

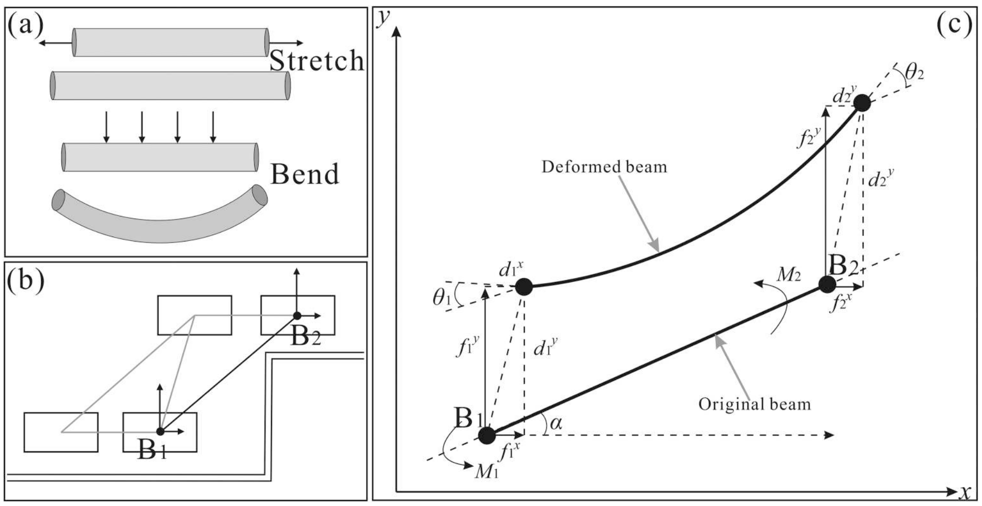


**Figure 2.** Illustration of the elastic beam algorithm (figure redrawn based on Liu et al. (2014)). (a) A single beam stretches or bends in response to external forces. (b) The spatial relationships between buildings are modeled as a proximity graph, with each edge treated as an elastic beam. The forces acting on buildings $B_1$ and $B_2$ from surrounding elements (e.g. roads) are shown as an example. (c) The beam connecting buildings $B_1$ and $B_2$ is deformed – either stretched or bent – under the applied forces.

Here, $A$, $I$, and $E$ are parameters that describe the material properties of the beams, which can be adjusted by the user. $A$ controls the stretching and compression behavior of the beam under axial load, $I$ governs its bending behavior, and $E$ adjusts the elasticity of the beams. $l$ is the length of the beam. While $c$ and $s$ are parameters determined by the angle ($\alpha$) between the beam axis and the x-axis, and $s = \sin\alpha$and $c = \cos\alpha$.

## 3. Methodology

The overall workflow of our approach is illustrated in Figure 3. First, labels are generated based on the current viewpoint and their corresponding point features. To enable the use of the elastic beam algorithm, these labels are organized into a proximity graph, which serves as the underlying elastic beam structure. Forces acting on this graph are then calculated according to the quality constraints defined in *Section* 2.1. The elastic beam algorithm is applied iteratively to adjust label positions based on these forces, and the process continues until the resulting layout meets the specified quality criteria. To improve computational efficiency, we adopt a clustering strategy inspired by previous work, grouping labels before applying the algorithm. The elastic beam algorithm is then executed independently within each cluster. The overall process is outlined as follows:

Step 1. **Initiation**. The initial labels are generated according to the viewpoints and their corresponding points.

Step 2. **Label clustering**. The labels are clustered into groups based on a minimum spanning tree-based approach.

Step 3. **Proximity graph construction**. Proximity graphs are constructed to represent the spatial relationships between the labels in each group.

Step 4. **Force computation**. Forces are calculated for the proximity graph based on the defined quality requirements in *Section* 2.1.

Step 5. **Using the elastic beam algorithm**. The elastic beam algorithm is applied to the proximity graph, assigning forces to adjust the labels' positions. If the adjusted label positions meet the defined quality requirements in *Section* 2.1, the process stops. Otherwise, return to Step 3 and repeat the process.

A detailed explanation of each step is provided in the following sections.

### *3.1. Step 1. Initiation*

In a 3D geographic scene, objects closer to the viewer appear larger, while those farther away appear smaller. Accordingly, label sizes also need to be dynamically adjusted based on the viewpoint and the distance to their associated point features. Let the set of point features to be labeled be denoted as $P = \{p_1, p_2, \ldots, p_m\}$. For any point $p_m \in P$, the font size $F_m$ of its label is determined by the distance $d_m$ between the viewpoint and $p_m$, along with the user-defined maximum font size $F_{\max}$. For the point closest to the viewpoint, denoted as $p_{\text{nearest}}$, the font size is set to $F_{\max}$, and $F_m$ is calculated using Equation (3).

$$F_m = \sqrt{\frac{d_{\text{nearest}}}{d_m}} \times F_{\max} \tag{3}$$

$d_{\text{nearest}}$ represents the distance from the viewpoint to $p_{\text{nearest}}$. To ensure readability, in line with the quality requirement $\boldsymbol{GR_2}$, the font size must not fall below a specified minimum threshold, $F_{\min}$. If the calculated

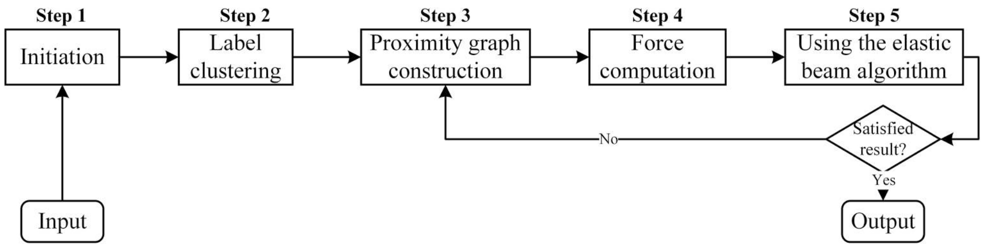


**Figure 3.** The framework.

font size $F_m$ for any point is less than $F_{\min}$, it is adjusted to $F_m = F_{\min}$. The initial generated label layout is shown in Figure 2, where the labels are represented by their bounding boxes.

### 3.2. Step 2. Label clustering

Labels to be displayed often cluster together, and the efficiency of the elastic beam algorithm is highly influenced by the number of elements involved (Liu et al. 2014; Wei et al. 2023). To improve computational efficiency, we group spatially adjacent labels into smaller clusters. For this purpose, we employ the commonly used Minimum Spanning Tree (MST) approach for clustering (Zhang, Ai, and Stoter 2013).

The label clustering process proceeds as follows: First, a Delaunay Triangulation (DT) is constructed using the center points of the labels to represent their proximity relationships, as shown in Figure 4(a). Based on this DT, Prim's algorithm is applied to generate an MST, with each edge weighted according to the nearest distance between two labels, as shown in Figure 4(b). For more details on Prim's algorithm, refer to Wei et al. (2018). Once the MST is constructed, long edges are sequentially removed to divide the labels into multiple subgroups. The stopping condition for this division process is determined by the desired number of labels $T_{\text{num}}$ per group. As illustrated in Figure 4(b), a long edge (highlighted in red) is removed from the MST to partition the label set into two distinct clusters.

### 3.3. Step 3. Proximity graph construction

Delaunay Triangulation (DT) is an effective method for representing the proximity relationships and spatial distribution of map features in graph form (Wei et al. 2018). In this study, we also apply DT to represent the spatial relationships among labels within a label group, as illustrated in Figure 5(a). However, if two labels connected by an edge in DT are obstructed by other labels, they are not considered to be in close proximity. Therefore, any edges in the DT that intersect other labels' bounding boxes are removed.

The process for constructing the proximity graph for labels within a group is as follows: First, a DT is generated using the center points of the labels, as illustrated in Figure 5(a). Next, edges within the DT

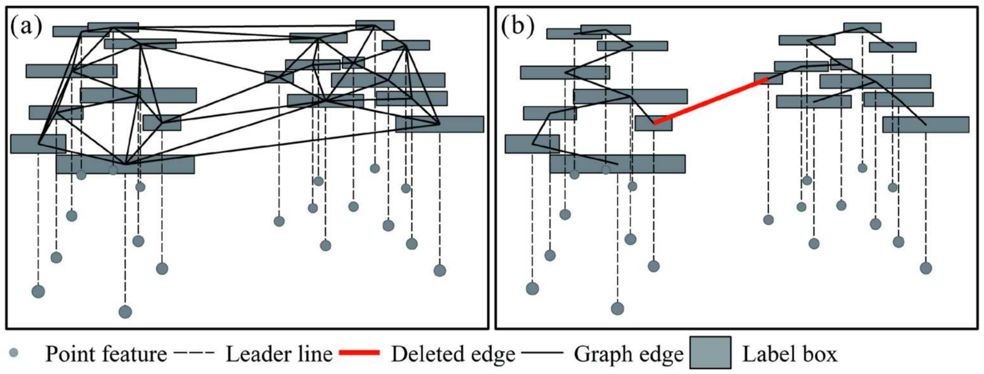


**Figure 4.** Label Clustering, the labels are represented by their bounding boxes. (a) A DT is constructed to represent the spatial and proximity relationships between buildings. (b) A MST is then generated based on the DT. Removing the long edge (shown in red) splits the labels into two clusters.

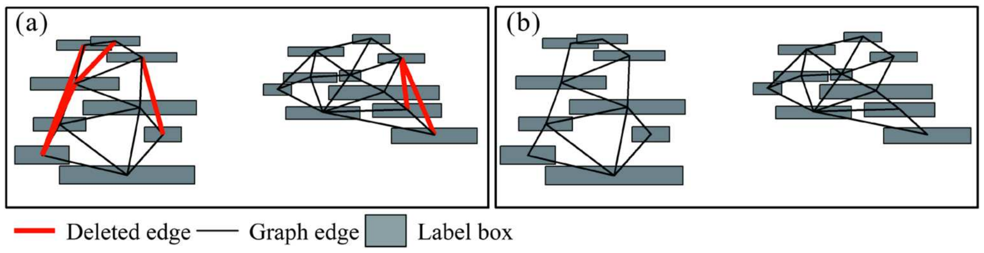


**Figure 5.** Proximity graph construction. (a) DTs are constructed for each group of labels. Edges those passing through other labels are removed from the DT, as indicated by the red edges. (b) The resulting proximity graphs after edge removal.

that intersect other labels are removed, as the red edges shown in Figure 5(a). The final proximity graph, with unnecessary edges removed, is depicted in Figure 5(b).

## 3.4. Step 4. Force computation

Four types of forces are computed for the built proximity graph ($G$): (1) The two labels connected by an edge in $G$ that overlap ($\boldsymbol{RR_1}$) or are closer than the minimum threshold distance ($d_{\min}$) ($\boldsymbol{RR_2}$) are repulsed by each other to avoid overlap or insufficient spacing; (2) Labels that overlap ($\boldsymbol{RR_1}$) or are within the minimum threshold distance ($d_{\min}$) ($\boldsymbol{RR_2}$) of an existing point feature are repulsed by the point feature to avoid overlap or insufficient spacing; (3) Labels that extend beyond the screen boundaries are repulsed by the screen to prevent them from going outside the visible area; (4) Labels that have been displaced horizontally by a large distance are subjected to an attractive force pulling them back towards their initial position to ensure the leader lines remain properly connected to the labels ($\boldsymbol{GR_1}$). Each node in $G$ may experience several forces, which need to be combined into a single resultant force.

## 3.5. Repulsive force from other labels

The two labels ($Label_i$ and $Label_j$) are repulsed away from each other if they have an overlap or are closer than the minimum threshold distance ($d_{\min}$). If $Label_i$ and $Label_j$ are closer than the minimum threshold distance ($d_{\min}$) but do not have an overlap, repulsive forces must be applied to ensure sufficient spacing between the two labels. The repulsive force ($ref_i$ and $ref_j$) on this occasion for $Label_i$ and $Label_j$ are defined as Equations (4), (5), and are shown in Figure 6(a).

$$f_i = 0.5 * \frac{\overrightarrow{p_j p_i}}{|\overrightarrow{p_j p_i}|} \times (d_{\min} - |\overrightarrow{p_j p_i}|) \tag{4}$$

$$f_j = 0.5 * \frac{\overrightarrow{p_i p_j}}{|\overrightarrow{p_i p_j}|} \times (d_{\min} - |\overrightarrow{p_i p_j}|) \tag{5}$$

Where $p_i$ and $p_j$ are the closest points on $Label_i$ and $Label_j$, respectively, as shown in Figure 6(a).

If $Label_i$ and $Label_j$ have an overlap, repulsive forces must be applied to ensure the overlap can be eliminated. Suppose the bounding rectangles of $Label_i$ and $Label_j$ are $[x^i_{\min}, y^i_{\min}, x^i_{\max}, y^i_{\max}]$, $[x^j_{\min}, y^j_{\min}, x^j_{\max}, y^j_{\max}]$, respectively, where $x_{\min}, y_{\min}, x_{\max}, y_{\max}$ denote the minimum and maximum values of the bounding rectangles in the $x$ and $y$ directions. Then the forces required to ensure the distance between labels is greater than $d_{\min}$ can be calculated separately in the four directions: $x$-, $x$+, $y$-, and $y$+, as defined in Equations (6)-(9) and are illustrated in Figure 6(b).

$$f_{x-} = x^i_{\max} - x^j_{\min} + d_{\min} \tag{6}$$

$$f_{x+} = x^j_{\max} - x^i_{\min} + d_{\min} \tag{7}$$

$$f_{y-} = y^i_{\max} - y^j_{\min} + d_{\min} \tag{8}$$

$$f_{y+} = y^j_{\max} - y^i_{\min} + d_{\min} \tag{9}$$

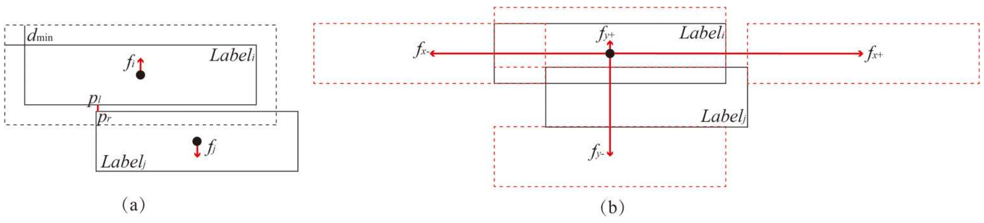


Figure 6. Repulsive force from other labels. (a) $Label_i$ and $Label_j$ are repulsed away from each other for they are closer than the minimum threshold distance ($d_{\min}$). (b) $Label_i$ and $Label_j$ are repulsed away from each other for they have an overlap.

Where $f_{x-}$, $f_{x+}$, $f_{y-}$, $f_{y+}$ are the potential forces in directions $x$-, $x$+, $y$-, and $y$+ to maintain sufficient spacing. The final force on labels $Label_i$ and $Label_j$ can then be determined based on the minimum force from these four directions, as $0.5 * \min(|fx-|, |fx+|, |fy-|, |fy+|)$. The force direction on $Label_i$ is the force of $\min(|fx-|, |fx+|, |fy-|, |fy+|)$, $Label_j$ moves in the opposite direction to $Label_i$.

### 3.5.1. Repulsive force from an existing point feature

If the distance between a label and an existing point feature is less than the threshold or there is an overlap, a repulsive force needs to be applied to the label to move it away from the feature, as shown in Figure 7. Similar to the repulsive forces applied to labels when they have an overlap, we also calculate the potential repulsive forces that may resolve the conflict in the four directions $x$-, $x$+, $y$- and $y$+, as defined in Equations (10)-(13).

$$f_{x-} = \frac{\overrightarrow{p_i p_{x-}}}{|\overrightarrow{p_i p_{x-}}|}(|\overrightarrow{p_i p_{x-}}| + d_{\min}) \tag{10}$$

$$f_{x+} = \frac{\overrightarrow{p_i p_{x+}}}{|\overrightarrow{p_i p_{x+}}|}(|\overrightarrow{p_i p_{x+}}| + d_{\min}) \tag{11}$$

$$f_{y-} = \frac{\overrightarrow{p_i p_{y-}}}{|\overrightarrow{p_i p_{y-}}|}(|\overrightarrow{p_i p_{y-}}| + d_{\min}) \tag{12}$$

$$f_{y+} = \frac{\overrightarrow{p_i p_{y+}}}{|\overrightarrow{p_i p_{y+}}|}(|\overrightarrow{p_i p_{y+}}| + d_{\min}) \tag{13}$$

Here, $p_i$ represents the point feature, and $p_{x-}$, $p_{x+}$, $p_{y-}$ and $p_{y+}$ are the projection points onto the bounding rectangle of the label in the four directions. The repulsive force on the label is then determined based on the minimum force from these four directions, as $\min(|fx-|, |fx+|, |fy-|, |fy+|)$. The force direction on the label is the force of $\min(|f_{x-}|, |f_{x+}|, |f_{y-}|, |f_{y+}|)$.

### 3.5.2. Repulsive force from the screen

Labels cannot extend beyond the screen boundaries. Therefore, if the distance between a label and the screen edge is less than the threshold distance $d_{\min}$, a repulsive force $f_{\text{screen}}$ is applied to the label to push it away from the screen, as defined in Equation (14).

$$f\,\text{screen} = \frac{\overrightarrow{p_s p_r}}{|\overrightarrow{p_s p_r}|} \times (d_{\min} - |\overrightarrow{p_s p_r}|) \tag{14}$$

Where $p_s$ and $p_l$ are the two closest points between the screen and the label.

### 3.5.3. Attractive force

If a label has been displaced too far horizontally, an attractive force $f_{\text{atr}}$ is applied to ensure the leader line remains connected to the label, the definition is shown as Equation (15).

$$f_{atr} = \overrightarrow{p'_i p'_{\text{nearest}}} \tag{15}$$

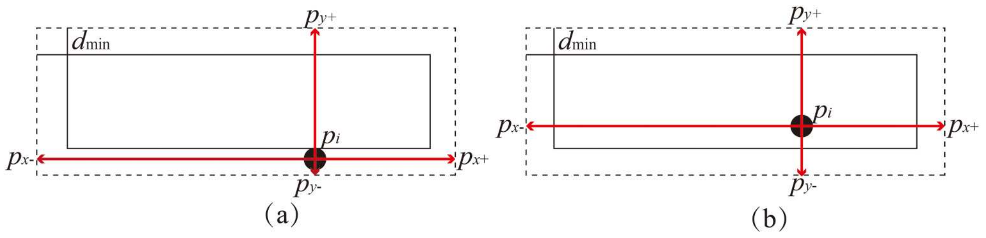


**Figure 7.** Repulsive force from from an existing point feature. (a) Label is repulsed away by an existing point feature $p_i$ for they are closer than the minimum threshold distance ($d_{\min}$). (b) Label is repulsed away by an existing point feature $p_i$ for they have an overlap.

Here, $p'_i$ represents the anchor point on the label for the leader line, and $p'_{\text{nearest}}$ is the closest point from $p'_i$ to the corresponding label.

### 3.5.4. Force combination

A label may be influenced by multiple external forces, either repulsive or attractive. Simply summing these forces using vector-based methods may lead to inappropriate displacement due to the cumulative effects of forces in similar directions. To address this, we follow the method proposed by Liu et al. (2014), which combines these forces by considering the local maximum component forces in four primary directions, for more details see Liu et al. (2014).

Furthermore, it is crucial to note that existing point features cannot be displaced. If a label is forced to move by two existing point features in opposite directions, it will always conflict with these features after displacement. In such cases, the force between the label and existing point features must be recalculated as follows: repulsive forces from different point features in the same direction (with an angle of ≤90°) are selected sequentially for combination. The resultant force with the smallest value after combination represents the force acting on the label from the existing point features.

## 3.6. Step 5. Using the elastic beam algorithm

The displacement of each node in the proximity graph ($G$) is computed by using the elastic beam algorithm after the construction of $G$ and the computation of the applied forces. The algorithm only once cannot guarantee that all conflicts that violate the quality requirements will be resolved or no new conflicts will arise, particularly in complex situations (Liu et al. 2014). Thus, an iterative strategy is applied until all labels are in proper locations. Two stop conditions are defined for the iterative process.

- **Condition 1**: The iterative process needs to stop if a satisfying result that satisfies the defined constraints in *Section* 2.1 is obtained after the displacement. It means the nodes don't need to be displaced and can be ruled by setting a threshold for the max($f_k$). If $\max(f_k) \le \varepsilon$then the iterative process stops, $\varepsilon$ is a small constant, i.e. $\varepsilon = 0.001$.
- **Condition 2**: The iterative process needs to stop if a maximum step ($T_s$) is reached.

The iterative process is implemented as **Algorithm 1**.

Algorithm 1. The iterative process for label displacement

```
Input: the points to be displayed as $P = \{p_1, p_2, \ldots, p_m\}$, the maximum step for the iterative process as $T_s$, the minimum distance threshold as $d_{\min}$.
Output: The label position after displacement.
Initiation Create the labels in default positions, their bounding boxes denoted as $R = \{r_1, r_2, \ldots, r_m\}$.
Divide the labels into groups, denoted as $\{C_1, \ldots, C_m\}$.
Foreach $C_m$ in $\{C_1, \ldots, C_m\}$
  The iterative step,$t \leftarrow 1$;
  Do
    Construction of the proximity graph ($G$);
    If $t \le T_s$ Then
      Force computation;
      Compute the displacement vector for each node in $G$ with the elastic beam algorithm, the maximum force as max($fk$);
      Update coordinates of each $rm \in R$ based on the displacement vector;
      $t \leftarrow t + 1$;
    While($f_k \le \varepsilon$ AND $t \le T_s$).
Return: $R = \{r_1, r_2, \ldots, r_m\}$
```

## 4. Experiments

### 4.1. Dataset

Two types of typical point features are used in the experiments. (1) Automatic identification System (AIS) data: A subset of open-source AIS data from January 1, 2020, covering the western coastal region of the

United States, was selected. This dataset, provided by the NOAA Office for Coastal Management, includes desensitized ship trajectory data for 60 vessels, with labels displayed in English (NOAA Office for Coastal Management). (2) Points of Interest (POI) data: POI data for the central urban area of Shanghai was obtained from MapBox. It includes 76 POIs, with labels displayed in Chinese.

### 4.2. Evaluation metric

We employed five metrics to assess the quality of label layout, adhering to the quality criteria specified in *Section* 2.1. These metrics include:

(1) Label-to-label conflict count ($N_{\text{r-r}}$): This metric evaluates overlap avoidance (***RR*₁**) and minimum distance maintenance ($\boldsymbol{RR_2}$). A conflict occurs when two labels are positioned closer than the specified minimum distance, $d_{\min}$, or overlap with each other.
(2) Label-to-feature conflict count ($N_{\text{r-p}}$): This metric evaluates overlap avoidance ($\boldsymbol{RR_1}$) and minimum distance maintenance ($\boldsymbol{RR_2}$). Here, a conflict is identified when a label is either within $d_{\min}$ of, or overlaps with, an existing map feature.
(3) Average deviation in relative direction between neighboring labels ($A_{\text{ms}}$): This metric evaluates the maintenance of relative relationships ($\boldsymbol{RR_3}$) (Wei et al. 2023). Neighboring labels are defined as pairs of labels connected by edges in the proximity graph established in *Section* 3.3. The $A_{\text{ms}}$ metric is calculated as shown in Equation (16).

$$A_{\text{ms}} = \frac{\sum_{m=1}^{N} \Delta O_m}{N} \tag{16}$$

Here, $N$ is the number of edges in the proximity graph, and $\Delta O_m$represents the directional deviation of the corresponding edge before and after displacement.

(1) Total displacement distance of all labels ($D_{\text{sum}}$): This metric evaluates the minimization of the overall displacement distance for all labels ($\boldsymbol{DR_1}$).
(2) Running time ($t$): This metric evaluates the efficiency of the algorithm.

### 4.3. Implement detail

#### 4.3.1. The approach for comparison

The GrowingBorder method is a widely used approach for point labeling in 3D scenes, and we adopt it here for comparison (Maass and Döllner 2006). Unlike the original version, which places labels starting from the screen border and generally produces labels of uniform size, we introduce refinements to make the method more adaptive to the 3D viewing context. Specifically, label positions are adjusted based on their distance from the viewpoint, and in each iteration, labels are placed with minimal displacement required to avoid conflicts with existing labels and point features. These modifications aim to produce more contextually appropriate and visually coherent layouts, enabling a fairer and more meaningful comparison.

#### 4.3.2. The parameter settings

The initial layout parameters for AIS and POI data labeling are set as follows: leader line length is 2 cm, the direction is 90°, and the maximum font size for labels is 12 pt. These parameters can be adjusted by users in practical applications. Based on visual perception guidelines (He, Song, and Li 2016), the minimum distance between labels, or between labels and map features, is set to $d_{\min} = 0.2$ mm. The $T_s$ for the iterative process of elastic beam algorithm application in *Section* 3.5 is set considering the number of the labels ($NumL$) according to Liu et al. (2014) and Wei et al. (2023) as: $T_s = 2*NumL$, and if $NumL < 10$, $NumL$ is set as 10.

### 4.3.3. Experiment environment

We implemented our proposed approach using JavaScript code. The experiments were conducted on a personal computer equipped with an AMD Ryzen 7-7840HS Radeon 780M Graphics @3.80 GHz CPU and 16GB RAM. The results are shown in Tables 1–3.

**Table 1.** The statistical analysis for label layouts.

| Data | Method | $N_{r\text{-}r}$ | $N_{r\text{-}p}$ | $A_{ms}$ | $D_{sum}$ | $T$ |
|---|---|---|---|---|---|---|
| AIS data | The initial layout | 26 | 15 | 0 | 0 | – |
| | The layout via refined GrowingBorder | 0 | 0 | 8.35 | 11.16 | 0.32s |
| | The layout via the proposed approach | 0 | 0 | 7.21 | 10.80 | 1.48s |
| POI data | The initial layout | 23 | 26 | 0 | 0 | – |
| | The layout via refined GrowingBorder | 0 | 0 | 11.99 | 12.60 | 0.37s |
| | The layout via the proposed approach | 0 | 0 | 9.95 | 14.36 | 1.52s |

**Table 2.** The three label layouts for POI data.

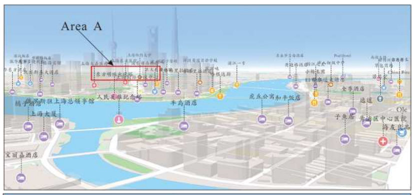
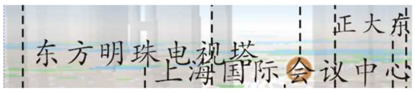
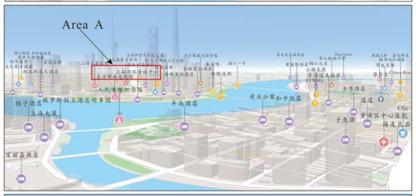
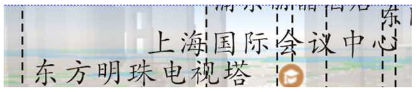
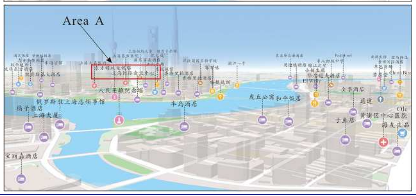
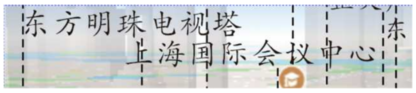
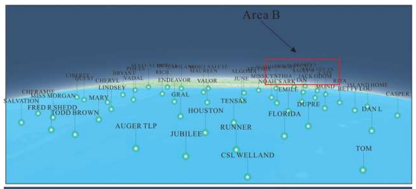
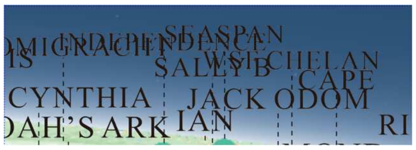
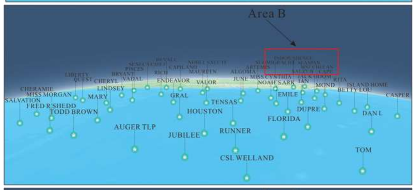
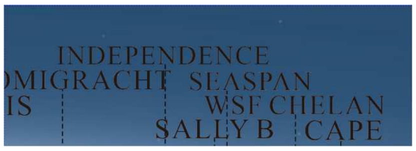
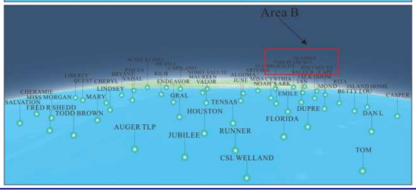
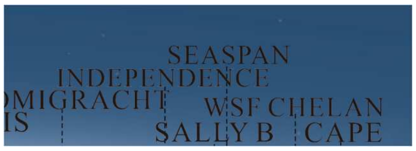
| Methods | The whole area | The enlarged area (Area A) |
|---|---|---|
| Initial layout |  |  |
| GrowingBorder |  |  |
| The proposed method |  |  |

**Table 3.** The three label layouts for AIS data.

| Methods | The whole area | The enlarged area (Area B) |
|---|---|---|
| Initial layout |  |   INDEPENDENCE SEASPAN SALLY B WSF CHELAN CAPE CYNTHIA JACK ODOM IAN |
| GrowingBorder |  |  |
| The proposed method |  |  |

### 4.4. Result

#### 4.4.1. Quantities analysis

The results in Table 1 show that in the absence of label placement approaches, conflicts are prominent in both datasets: AIS data has 26 label-to-label conflicts and 15 label-to-feature conflicts, while POI data has 23 and 26 conflicts, respectively. Both the proposed method and the refined GrowingBorder approach effectively resolve all such conflicts, as evidenced by the elimination of $N_{\text{r-r}}$ and $N_{\text{r-p}}$ conflicts post-optimization. This indicates that both methods successfully prevent label overlaps and ensure minimum required distances. However, repositioning labels can impact their relative spatial relationships. As indicated in Table 1, the proposed method results in a smaller average deviation in relative direction ($A_{\text{ms}}$) for both datasets: 7.21 for AIS and 9.95 for POI, as compared to GrowingBorder's 8.35 and 11.99, respectively. This suggests that the proposed method better preserves relative positioning, likely due to its use of a global neighborhood control that adjusts labels collectively. Despite these advantages in preserving relative relationships, the proposed method incurs a higher computational cost due to its global optimization strategy. For AIS and POI datasets, the algorithm takes 1.48s and 1.52s, respectively, as opposed to GrowingBorder's faster times of 0.32s and 0.37s. Additionally, the total displacement distance ($D_{\text{sum}}$) varies between the methods. In the AIS dataset, the proposed method yields a $D_{\text{sum}}$ of 10.80, which is slightly smaller than GrowingBorder's 11.16, whereas in the POI dataset, the proposed method's $D_{\text{sum}}$ of 14.36 is higher than GrowingBorder's 12.60. These results indicate that although neither method consistently minimizes total displacement, the proposed method excels in preserving label spatial relationships, albeit with increased computational time.

#### 4.4.2. Qualitative analysis

Tables 2 and 3 illustrate different label layouts generated using various approaches for the POI and AIS datasets. As shown in Tables 2 and 3, when no label placement strategies are applied, conflicts are prominent in both datasets. However, as observed in Tables 2 and 3, both the proposed approach and the refined GrowingBorder effectively avoid these conflicts. Comparing the proposed approach with the refined GrowingBorder, it is evident that the proposed method better preserves relative spatial relationships between labels, as shown in Area A and Area B. For example, in Area A, the labels for '东方明珠电视塔' (Oriental Pearl TV Tower) and '上海国际会议中心' (Shanghai International Convention Center) retain their relative orientation under the proposed approach. Similarly, in Area B, the labels 'INDEPENDENCE' and 'SEASPAN' maintain their relative orientation. In contrast, these spatial relationships are disrupted in layouts generated by the refined GrowingBorder. However, to preserve relative orientations, the proposed method may also result in greater label displacement, as observed in Areas A and B. This suggests that while the proposed approach excels in maintaining spatial relationships, it may lead to an increase in total label displacement distance.

In summary, the experimental results highlight a clear trade-off between spatial consistency and computational efficiency. Compared with the refined GrowingBorder method, our approach demonstrates stronger performance in preserving the relative spatial relationships between labels, as reflected in the lower average directional deviation ($A_{\text{ms}}$). This advantage stems from the global optimization mechanism of the elastic beam algorithm, which adjusts labels collectively to maintain spatial coherence. However, this benefit comes at the cost of longer runtime. Since the elastic beam algorithm solves the displacement problem iteratively over a proximity graph, it requires more computation, particularly for large label sets. In real-world applications, this trade-off is context-dependent. For example, in offline scenarios such as producing static 3D maps or labeling high-resolution images, the extra computation is acceptable in exchange for improved label consistency and clarity. On the other hand, for interactive or real-time applications – such as dynamic visualization in mobile AR or online navigation – users may prefer faster methods with slightly less spatial precision. Our method is thus best suited for high-precision labeling tasks where preserving spatial relationships is critical, and it is not designed for frame-by-frame recomputation under high-frequency viewpoint changes. Nevertheless, applying our approach in slowly evolving scenes, such as geographic videos or visual narratives where the 3D environment changes gradually, remains a promising direction. In such contexts, multi-frame incremental computation becomes both feasible and beneficial. The spatial consistency advantages of our method can still be retained across adjacent frames. However, to fully support these use cases, additional challenges – such as temporal stability and inter-frame layout coherence – must be addressed.

## 5. Extension

### 5.1. Application for other types of labels

In practical applications, the type of label leader line often needs to be adjusted to meet specific user requirements. This study employs a widely used approach in which the leader line direction is fixed, but the anchor point connecting the leader line to the label is flexible. However, some users may require a fixed anchor point, and our approach can also be adapted to accommodate this option. For instance, by setting the leader line direction to 90° and fixing the anchor point to the center of the label, the label is constrained to vertical movement only. As a result, if the label extends beyond the screen edge horizontally, simply repositioning the label is insufficient to prevent conflicts, necessitating the deletion of both the label and its associated feature, as illustrated with the label '文华东方酒店' (Mandarin Oriental Hotel) in Table 3. Additionally, when calculating forces, the forces on the label need to be decomposed along the primary axis of the leader line, retaining only the forces in the fixed direction. During the iterative solution process of the Beams algorithm, the label's position update must also decompose the displacement vector and move the label only in the fixed direction. Using this setup, label placement for POI data was configured, with results shown in Figure 8.

As demonstrated in Figure 8, this method can configure label placement based on fixed leader line direction and anchor point settings, resulting in conflict-free layouts. Comparing Table 3 and Figure 8, it is evident that the fixed anchor point in Figure 8 restricts the labels' movement space. Consequently, the labels must travel greater distances to find suitable positions, resulting in a larger total displacement (increased by 4.11) and a slightly higher deviation in relative direction between neighboring labels (an increase of 0.13°). For practical applications where users aim to maintain minimal changes in displacement distance and relative direction between neighboring labels, it is recommended to use configurations where the leader line direction is fixed, but the anchor point remains flexible. Conversely, if stability in relative direction is less critical, a fixed leader line direction with a fixed anchor point may be preferable.

### 5.2. Application for label placement in images

Although the label placement algorithm in this study is designed for 3D geographic scenes, it is ultimately displayed on a 2D screen view based on the 3D scene, enabling label placement in a 2D visual context. Therefore, this method can also be adapted for label placement in certain image-based contexts, such as panoramic images, where it has already found numerous applications (Gemsa, Haunert, and Nöllenburg

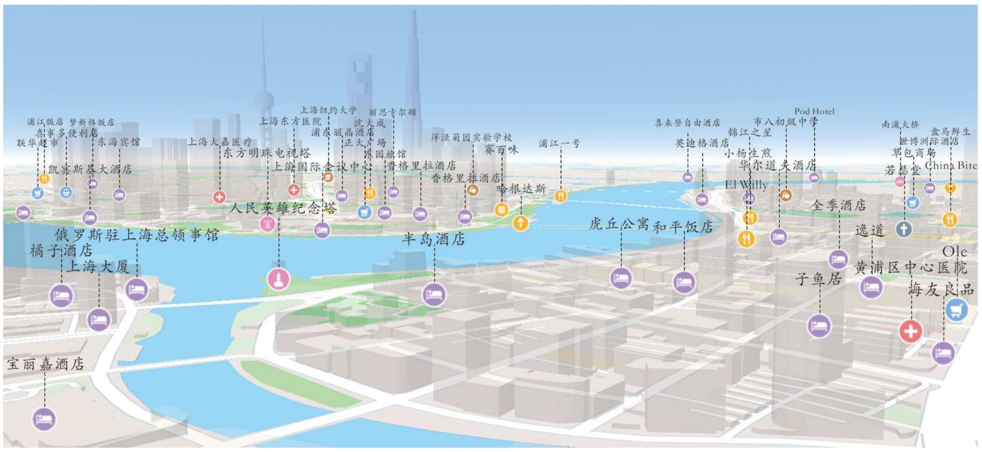


Figure 8. Label layouts for POI data with leader lines fixed in direction and anchor points fixed to the bottom center of each label.

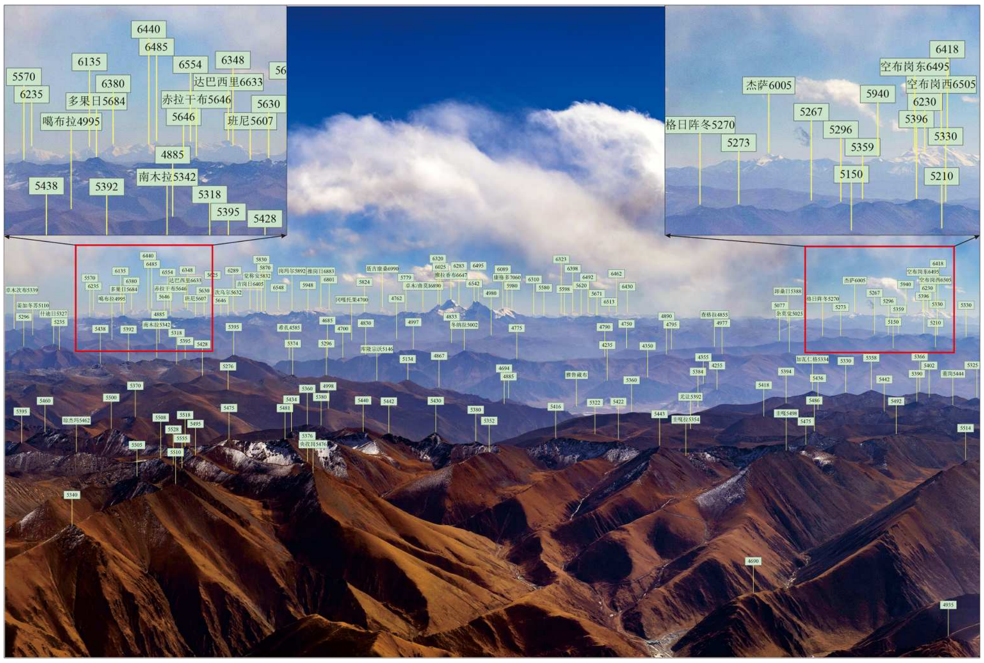


**Figure 9.** The label layouts for the image (part of the Himalayas).

2015). In collaboration with a National Geographic photographer, we applied the method to label mountain names and elevations in a map of part of the Himalayas (Kangtuo Mountain Range, Kangphu Gang Range, eastern section of the Lagoi Kangri Mountains, and the Gyala Peri Mountain Range). But unlike the labels in a 3D geographic scene, the labels in images are usually kept the same size. The results are shown in Figure 9. As illustrated in Figure 9, this method effectively enables label placement for features in image-based contexts as well.

## 6. Parameter analysis

### 6.1. The influence of the cluster size

To enhance algorithm efficiency, this study employs a clustering approach dividing the label group into multiple subgroups. The parameter $T_{\text{num}}$ controls the size of each subgroup. Since the complexity of the Beams algorithm increases exponentially with the number of elements, different subgroup divisions can impact performance (Liu et al. 2014). To evaluate this, we set $T_{\text{num}}$ values of 10, 20, 30 and $+\infty$for the AIS dataset, analyzing the effect of varying cluster sizes. The results are summarized in Table 4.

As Table 4 indicates, dividing labels into subgroups significantly improves algorithm efficiency and reduces total label displacement distance, with smaller subgroups leading to greater efficiency and reduction in displacement. When $T_{\text{num}}$ is set to 10, 20, 30, and $+\infty$, the algorithm runtime is 0.43s, 1.48s, 3.64s, and 6.28s, respectively, and the total label displacement distance is 10.54, 10.80, 10.86, and 10.90. However, as subgroups are further divided, the average directional deviation between neighboring labels increases. Furthermore, dividing the labels into small groups may still fail to resolve all label conflicts. For example, if $T_{\text{num}}$ is set to 10, two label conflicts cannot be resolved. The reason is that conflicts may arise between labels in different groups and post-procedure is required. In practice, selecting an optimal subgroup size is crucial to balancing efficiency with effectiveness, and a $T_{\text{num}}$ of 20 appears to be a reasonable compromise.

**Table 4.** The efficiency analysis.

| | $N_{\text{r-r}}$ | $N_{\text{r-}p}$ | $A_{\text{ms}}$ | $D_{\text{sum}}$ | $t/s$ |
|---|---|---|---|---|---|
| $T_{\text{num}} = 10$ | 2 | 0 | 7.73 | 10.54 | 0.43 |
| $T_{\text{num}} = 20$ | 0 | 0 | 7.21 | 10.80 | 1.48 |
| $T_{\text{num}} = 30$ | 0 | 0 | 7.14 | 10.86 | 3.64 |
| $T\text{num} = \infty$ | 0 | 0 | 7.09 | 10.90 | 6.28 |

**Table 5.** The influence analysis of the proximity graphs.

| | $N_{r\text{-}r}$ | $N_{r\text{-}p}$ | $A_{ms}$ | $D_{sum}$ | $t/s$ |
|---|---|---|---|---|---|
| MST | 0 | 0 | 7.31 | 10.71 | 1.39 |
| DT | 0 | 0 | 7.21 | 10.80 | 1.48 |

### *6.2. The influence of the proximity graphs*

In this study, an adjusted Delaunay Triangulation (DT) is used to represent spatial relationships between labels, while other proximity graphs such as MST is also commonly employed to describe spatial relationships between map features, and MST $\subseteq$ DT(Wei et al. 2018). Therefore, we also use MST to express spatial relationships between labels and apply the proposed method to position the labels. The influences of different proximity graphs on the results are assessed according to the evaluation metrics and are shown in Table 5. Table 5 indicates that using MST to represent spatial relationships between labels can also produce acceptable results, ensuring no conflicts exist between labels or between labels and existing map features. Additionally, since MST is a subset of DT, constraining fewer proximity relationships, the runtime $t$ is reduced by 0.09 s, resulting in higher efficiency. Furthermore, the total displacement distance significantly decreases, reducing by 0.09(0.8%). However, since MST constrains fewer proximity relationships, the relative directional deviation between neighboring labels increases by 0.1(1.4%). In practical applications, users seeking greater efficiency may opt for MST to construct the proximity graph, while those aiming to better maintain directional relationships between labels may prefer DT.

## 7. Conclusion

To effectively achieve the placement of leadered point feature labels in a three-dimensional (3D) geographic scene, this paper models the problem as a map surface target displacement issue for a two-dimensional (2D) screen display. It applies the classical Elastic Beams Displacement Method to iteratively calculate the label layout that satisfies the placement constraints. Given that the Elastic Beams Displacement Method utilizes a proximity graph to represent the spatial relationships between labels and employs this graph as an overall framework to control changes in spatial relationships during the label displacement process, the method proposed in this paper can better maintain the relative relationships between labels. The experimental results effectively confirm this conclusion and demonstrate the applicability of our method to image labeling. However, while our approach aims for global optimization, it does come with increased computational costs and may result in greater total label displacement distances. Future research will explore efficient strategies that integrate spatial indexing and develop timely label placement methods that comprehensively consider user interaction processes, as well as extend the current approach to gradually changing 3D scenes such as geographic videos, where temporal stability and inter-frame coherence become essential.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This work was supported by grants from the National Natural Science Foundation of China (No. 42501551, No. 42371455 and No. 42171438).

## Data and code availability statement and data deposition

Upon request, the data and code will be duly provided.

## ORCID

*Zhiwei Wei* http://orcid.org/0000-0002-3494-3686

## References

Araujo, E., A. Chaves, and L. Lorena. 2019. "Improving the Clustering Search Heuristic: An Application to Cartographic Labeling." *Applied Soft Computing* 77:261–273. https://doi.org/10.1016/j.asoc.2018.11.003.

Bader, M. 2001. "Energy Minimization Methods for Feature Displacement in map Generalization." PhD diss., University of Zurich. https://geo.uzh.ch/dam/jcr:2c2e14d7-7754-41fe-81b6-fbd6c75b2aed/thesis_MatsBader_2001.pdf

Bobák, P., L. Čmolík, and M. Čadík. 2020. "Temporally Stable Boundary Labeling for Interactive and non-interactive Dynamic Scenes." *Computers & Graphics* 91:265–278. https://doi.org/10.1016/j.cag.2020.08.005.

Bobák, P., L. Čmolík, and M. Čadík. 2023. "Reinforced Labels: Multi-agent Deep Reinforcement Learning for Point-Feature Label Placement." *IEEE Transactions on Visualization and Computer Graphics* 30 (9): 5908–5922. https://doi.org/10.1109/TVCG.2023.3313729.

Cao, W., F. Peng, X. Tong, H. Dai, and Y. Zhang. 2022. "A Point-Feature Label Placement Algorithm considering Spatial Distribution and Label Correlation." *Acta Geodaetica et Cartographica Sinica* 51 (2): 301–311. https://doi.org/10.11947/j.AGCS.2022.20210247.

Chen, C., L. Zhang, J. Ma, Z. Kang, L. Liu, and X. Xue. 2010. "Adaptive Multi-resolution Labeling in Virtual Landscapes." *International Journal of Geographical Information Science* 24 (6): 949–964. https://doi.org/10.1080/13658810903473205.

Christensen, J., J. Marks, and S. Shieber. 1995. "An Empirical Study of Algorithms for Point-Feature Label Placement." *ACM Transactions on Graphics (TOG)* 14 (3): 203–232. https://doi.org/10.1145/212332.212334.

Dijk, S. V., M. V. Kreveld, T. Strijk, and A. Wolff. 2002. "Towards an Evaluation of Quality for Names Placement Methods." *International Journal of Geographical Information Science* 16 (7): 641–661. https://doi.org/10.1080/13658810210138742.

Dong, W., T. Yang, H. Liao, and L. Meng. 2020. "How Does map use Differ in Virtual Reality and Desktop-Based Environments?" *International Journal of Digital Earth* 13 (12): 1484–1503. https://doi.org/10.1080/17538947.2020.1731617.

Fan, H., Z. Zhang, and D. Du. 2005. "Quality Evaluation Model for map Labeling." *Geo-Spatial Information Science* 8 (1): 72–78. https://doi.org/10.1007/BF02826996.

Fan, Y., S. Lv, J. Guo, and F. Zhou. 2023. "A Survey of Label Placements in Visualizations." *Journal of Computer-Aided Design & Computer Graphics* 35 (08): 1162–1174. https://doi.org/10.3724/SP.J.1089.2023.19913.

Gemsa, A., J. H. Haunert, and M. Nöllenburg. 2015. "Multirow Boundary-Labeling Algorithms for Panorama Images." *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 1 (1): 1–30. https://doi.org/10.1145/2794299.

Gomes, S. P., L. A. N. Lorena, and G. M. Ribeiro. 2016. "A Constructive Genetic Algorithm for Discrete Dispersion on Point Feature Cartographic Label Placement Problems." *Geographical Analysis* 48 (1): 43–58. https://doi.org/10.1111/gean.12082.

Hao, T., and Z. Lin. 2022. "Optimal 3D Indoor Space Annotation Allocation Method." *Geo-Spatial Information* 20 (1): 27–31+51+6-7.

He, Z., Y. Song, and L. Li. 2016. *Cartography*. Wuhan, People's Republic of China: Wuhan University Press.

Lessani, M. N., J. Deng, and Z. Guo. 2021. "A Novel Parallel Algorithm with Map Segmentation for Multiple Geographical Feature Label Placement Problem." *ISPRS International Journal of Geo-Information* 10 (12): 826. https://doi.org/10.3390/ijgi10120826.

Lessani, M. N., Z. Li, J. Deng, and Z. Guo. 2025. "An MPI-Based Parallel Genetic Algorithm for Multiple Geographical Feature Label Placement Based on the Hybrid of Fixed-Sliding Models." *Geo-spatial Information Science* 28 (2): 761–779. https://doi.org/10.1080/10095020.2024.2313326.

Li, L., H. Zhang, H. Zhu, and W. Hu. 2018. "A Point-Feature Labeling Algorithm Based on Movable Regions." *Geomatics and Information Science of Wuhan University* 43 (8): 1129–1137. https://doi.org/10.13203/j.whugis20160289.

Liu, Y., Q. Guo, Y. Sun, and X. Ma. 2014. "A Combined Approach to Cartographic Displacement for Buildings Based on Skeleton and Improved Elastic Beam Algorithm." *PLoS One* 9 (12): e113953. https://doi.org/10.1371/journal.pone.0113953.

Lu, F., J. Deng, S. Li, and H. Deng. 2019. "A Hybrid of Differential Evolution and Genetic Algorithm for the Multiple Geographical Feature Label Placement Problem." *ISPRS International Journal of Geo-Information* 8 (5): 237. https://doi.org/10.3390/ijgi8050237.

Maass, S., and J. Döllner. 2006. "Efficient View Management for Dynamic Annotation Placement in Virtual Landscapes." Smart Graphics: 6th International Symposium, SG 2006, Vancouver, Canada, July 23-25, 2006. Proceedings 6 (pp. 1-12). Springer Berlin Heidelberg.

NOAA Office for Coastal Management. "AIS Data for 2020." NOAA. Accessed June 11, 2021. https://coast.noaa.gov/htdata/CMSP/AISDataHandler/2020/index.html.

Peng, S., Y. Song, and F. Wu. 2007. "The Research of Intelligent Point-Feature Cartographic Label Placement Base on ant Colony Algorithm." *Science of Surveying and Mapping* 32 (5): 80–81. https://doi.org/10.3771/j.issn.1009-2307.2007.05.029.

She, J., X. Li, J. Liu, Y. Chen, J. Tan, and G. Wu. 2019. "A Building Label Placement Method for 3D Visualizations Based on Candidate Label Evaluation and Selection." *International Journal of Geographical Information Science* 33 (10): 2033–2054. https://doi.org/10.1080/13658816.2019.1606431.

Su, S., L. Wang, Q. Du, J. Zhang, M. Kang, and M. Weng. 2023. "Revisiting Narrative Maps: Fundamental Theoretical Issues and a Research Agenda." *Acta Geodaetica et Cartographica Sinica* 52 (12): 2178–2196. https://doi.org/10.11947/j.AGCS.2023.20220324.

Tatzgern, M., D. Kalkofen, R. Grasset, and D. Schmalstieg. 2014. "Hedgehog Labeling: View Management Techniques for External Labels in 3D Space." 2014 IEEE Virtual Reality (VR), Minneapolis, USA.

van Kreveld, M., T. Strijk, and A. Wolff. 1998. "Point set Labeling with Sliding Labels." Proceedings of the Fourteenth Annual Symposium on Computational Geometry, 337–346, Minneapolis, USA, June 1998.

Wei, Z., S. Ding, W. Xu, L. Cheng, S. Zhang, and Y. Wang. 2023. "Elastic Beam Algorithm for Generating Circular Cartograms." *Cartography and Geographic Information Science* 50 (4): 371–384. https://doi.org/10.1080/15230406.2023.2196732.

Wei, Z., Q. Guo, L. Wang, and F. Yan. 2018. "On the Spatial Distribution of Buildings for map Generalization." *Cartography and Geographic Information Science* 45 (6): 539–555. https://doi.org/10.1080/15230406.2018.1433068.

Wei, Z., J. Shi, N. Yang, Y. Zhang, W. Xu, S. Ding, M. Li, and R. Guo. 2025a. "Multi-Row Labeling with Semantic Analysis: A Case Study on Chinese POIs." *Transactions in GIS* 29 (2): e70024. https://doi.org/10.1111/tgis.70024.

Wei, Z., N. Yang, W. Xu, S. Ding, M. Li, Y. Li, and R. Guo. 2025b. "Optimized 3D Point Labeling with Leaders Using the Beams Displacement Method." *Journal of Computer-Aided Design & Computer Graphics*, https://doi.org/10.3724/SP.J.1089.2024-00619.

Wolff, A., and T. Strijk. 1999. "The Map-Labeling Bibliography." http://i11www.iti.uni-karlsruhe.de/map-labeling/bibliograph.

Zhang, X., T. Ai, and J. Stoter. 2013. "Building Pattern Recognition in Topographic Data: Examples on Collinear and Curvilinear Alignments." *Geoinformatica* 17 (1): 1–33. https://doi.org/10.1007/s10707-011-0146-3.

Zheng, C., Q. Guo, and X. Liu. 2006. "Automatic Placement of Point Annotation Based on Tabu Search." *Geomatics and Information Science of Wuhan University* 31 (5): 428–431. https://doi.org/10.3321/j.issn:1671-8860.2006.05.013.

Zhou, X., Z. Sun, C. Wu, and Y. Ding. 2015. "Automatic Label Placement of Point Feature: Using ant Colony Algorithm Based on Group Clustering." *Journal of Geo-Information Science* 17 (8): 902–908. https://doi.org/10.3724/SP.J.1047.2015.00902.

Zhou, X., C. Wu, Z. Sun, Y. Ding, and T. He. 2016. "A 3D Annotation Optimal Placement Algorithm for the Point Features in the Small Scale Geographic Scene." *Acta Geodaetica et Cartographica Sinica* 45 (12): 1476–1484. https://doi.org/10.11947/j.AGCS.2016.20160210.

Zoraster, S. 1997. "Practical Results Using Simulated Annealing for Point Feature Label Placement." *Cartography and Geographic Information Systems* 24 (4): 228–238. https://doi.org/10.1559/152304097782439259.