Overlap-Free Labeling of Clustered Networks Based on Voronoi Tessellation[☆]

Hsiang-Yun Wu[a,*], Shigeo Takahashi[b], Rie Ishida[c]

[a] *The Institute of Computer Graphics and Algorithms, TU Wien, Vienna 1040, Austria*
[b] *School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, 965-8580 Japan*
[c] *Graduate School of Frontier Sciences, University of Tokyo, Kashiwa, 277-8561 Japan*

Abstract

Properly drawing clustered networks significantly improves the visual readability of the meaningful structures hidden behind the associated abstract relationships. Nonetheless, we often degrade the visual quality of such clustered graphs when we try to annotate the network nodes with text labels due to their unwanted mutual overlap. In this paper, we present an approach for aesthetically sparing labeling space around nodes of clustered networks by introducing a space partitioning technique. The key idea of our approach is to adaptively blend an aesthetic network layout based on conventional criteria with that obtained through centroidal Voronoi tessellation. Our technical contribution lies in choosing a specific distance metric in order to respect the aspect ratios of rectangular labels, together with a new scheme for adaptively exploring the proper balance between the two network layouts around each node. Centrality-based clustering is also incorporated into our approach in order to elucidate the underlying hierarchical structure embedded in the given network data, which also allows for the manual design of its overall layout according to visual requirements and preferences. The accompanying experimental results demonstrate that our approach can effectively mitigate visual clutter caused by the label overlaps in several important types of networks.

Keywords: Adaptive layout blending, Force-directed layouts, Centroidal Voronoi tessellation, Chebyshev distance, Centrality-based clustering

1. Introduction

Networks can successfully capture meaningful features hidden behind abstract data, such as social friendships, research co-authorships, co-purchasing relationships, etc. They represent such abstract data in a visually plausible manner by schematizing individuals/entities as the nodes and their relationships as the edges. Thus, visualizing the associated networks is considered a promising approach to enhancing the readability of the underlying abstract relationships.

Many important techniques have been developed for aesthetically laying out the networks on the screen space while avoiding unwanted node overlaps and edge crossings. The force-directed approach is one such representative technique; it applies repulsive and attractive forces to the nodes and edges of the network and then seeks the finalized network layout as its equilibrium state. Nonetheless, once we try to annotate the network nodes with text labels, we inevitably struggle against distracting visual clutter arising from the occlusions of such annotation labels due to the lack of labeling space around the respective nodes. This becomes even worse when handling densely connected networks. Incorporating space-partitioning techniques may mitigate such visual clutter, although it incurs edge overlaps as another problem and often degrades the networks' original aesthetic layouts.

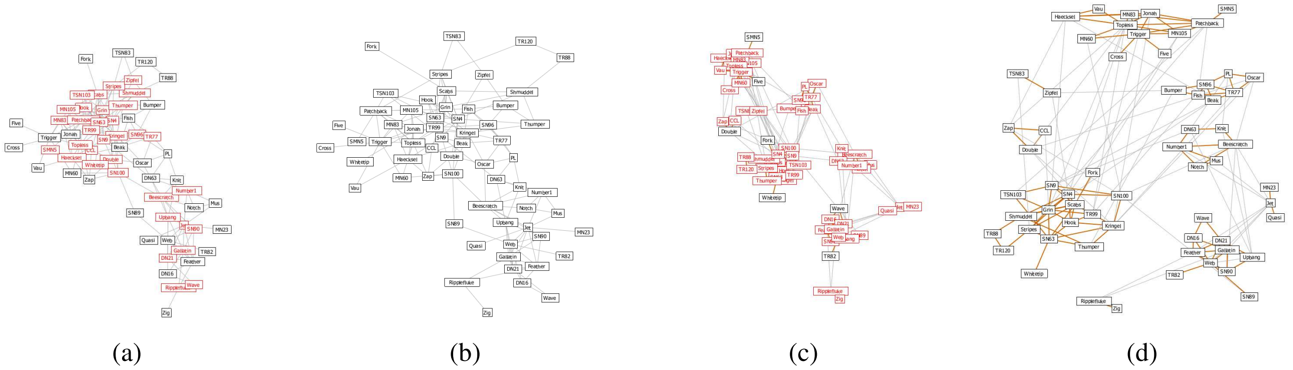
This is also true for clustered networks that reflect small communities inherent in the given abstract data. In fact, it is often the case that we first identify a set of representative small clusters in each network and rearrange its layout to specifically visualize such clusters. Labeling the nodes of such hierarchical networks while maintaining the corresponding structural layouts will be useful, in the same way as that for the aforementioned ordinary networks, for ensuring better readability of the underlying characteristics of the network data.

This paper presents an approach to overlap-free network labeling by sparing sufficient labeling space around the network nodes [1, 2]. This is accomplished by seeking an adaptive compromise between the layouts obtained through the conventional force-directed approach and through centroidal Voronoi tessellation. An anisotropic Chebyshev distance metric is incorporated into the Voronoi tessellation in order to fully respect the rectangular shapes of text labels associated with the network nodes. Furthermore, as an optional preprocess, centrality-based network analysis is introduced so that we can extract representative small clusters in the network to better reflect the underlying structures into the network layout. An extension of the aforementioned overlap-free labeling to this class of clustered networks will be formulated to maximally retain such hierarchical layouts. Experimental results are provided to demonstrate how effectively the proposed approach can annotate several typ-

[☆] A preliminary version of this work was presented at the 20th International Conference on Information Visualisation, July 19-22, 2016, Lisbon, Portugal.

[*] Corresponding author

Email addresses: hsiang.yun.wu@acm.org (Hsiang-Yun Wu), takahashis@acm.org (Shigeo Takahashi), ishidar92@gmail.com (Rie Ishida)



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

Figure 1: Network layouts of social relationships between dolphins [3]. (a) A simple force-directed network layout and (b) the corresponding layout obtained by our approach. (c) A clustered version of the force-directed network layout and (d) the corresponding layout obtained by our approach. Annotation labels in red have overlaps with others. Edges in orange indicate node connections contained in the same cluster obtained by the centrality-based network decomposition.

ical types of networks.

Figure 1 demonstrates that our approach can annotate network nodes with text labels without unwanted overlaps, even for an ordinary network and its clustered version. By simply applying the conventional force-directed approach, we cannot fully avoid overlaps between the annotation labels (Figure 1(a)). By contrast, our approach, with the help of space partitioning, can successfully spare enough labeling space around the respective nodes (Figure 1(b)). The clustered version is more likely to have mutual overlaps between the node labels due to the space limitation (Figure 1(c)). Even in this case, our approach allows us to effectively rearrange the node labels while retaining the original clustered structure for better readability of the corresponding abstract relationships (Figure 1(d)).

The remainder of this paper is organized as follows. First, we conduct a brief survey of related work in Section 2. In Section 3, we then present our idea of blending network layouts with that obtained by space partitioning in order to mitigate label overlaps. The enhancement of distance metrics in space partitioning to fully respect the rectangular shapes of the text labels is covered in Section 4. We describe the details of our algorithm for adaptively blending the two network layouts by individually controlling the blending ratio associated with each network node in Section 5. In Section 6, we extend this technical formulation for simple networks to clustered networks, basing our approach on the centrality-based network decomposition for inferring underlying hierarchical structures. After applying our approach to several typical types of networks as experimental results in Section 7, we conclude this paper in Section 8.

2. Related Work

In this section, we review the research most relevant to this study and classify it into two categories: network visualization and label annotation.

2.1. Network visualization

Network visualization has been intensively investigated due to its technical importance. In practice, it facilitates our understanding of the complicated relationships embedded in abstract network data, including social friendship between individuals, co-authorships between scientific researchers, co-purchase links on Internet marketing, etc [4]. Improving network readability, as well as reducing its visual complexity, has been an important research challenge, and several advanced network visualization techniques have been developed for this purpose.

As the pioneering physically based graph drawing algorithms, the force-directed approach [5, 6] was developed to simulate dynamics of network behavior and is widely used to arrange aesthetic layouts of networks. The associated algorithms usually model the network nodes as particles with electronic charges and edges as coil springs, in order to optimize the network layout as the equilibrium state of attractive and repulsive forces arising from Hooke's law and Coulomb's law [7]. This scheme maximally avoids unequal distribution of nodes and unwanted crossings between the edges, and thus naturally leads to an aesthetic layout of the network from a physical point of view.

The physical model of the conventional force-directed approach was improved by Hu [8], who introduced multilevel representation of the network based on octrees in search for its optimal layout. Simonetto et al. [9] accelerated the computation of the model to realize interactive control of aesthetic criteria for network visualization. The aesthetic aspects of network layouts were empirically investigated through the analysis of human drawings for networks [10] and their clustered versions [11]. Eye-tracking experiments were conducted to explore the aesthetic criteria of the network layouts [12]. Additional criteria were incorporated to visually discriminate cluster structures embedded in networks [13, 14, 15] and their 3D representations [16].

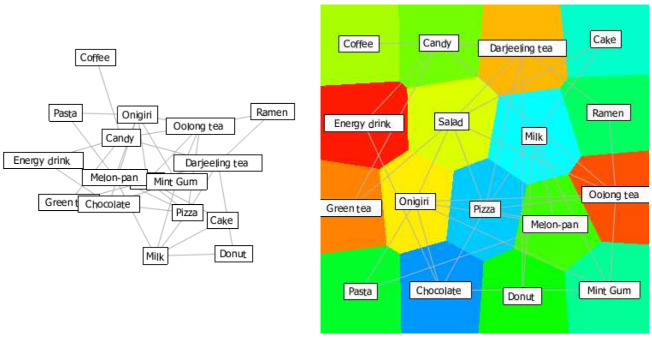
2.2. Label annotation

Although the force-directed algorithms provide visually pleasing layouts of network nodes in general, they cannot explicitly exclude mutual overlaps between text labels associated with the network nodes. To directly solve this problem, techniques were further invented that explicitly control the available space around the network nodes so that we can insert rectangular annotation labels. Among these techniques, Dwyer et al. [17] maximized space coverage of the screen space by rearranging the annotation labels while maintaining the relative positions of nodes with respect to the horizontal and vertical directions. Another relevant approach was proposed by Gansner et al. [18], who extended the aforementioned physical model by introducing a stress factor to generate sufficient labeling space around each node. Note that these approaches can be categorized as postprocessing approaches that faithfully respect the original network layouts specified by users or derived from available layout algorithms. The D3 library [19], which is well-known advanced tool for information visualization, enables us to detect collisions between network node labels, although it does not completely maintain their relative positions.

Moreover, space-partitioning techniques such as Voronoi tessellation have been introduced to properly prepare space for annotation labels. They help us improve the allocation of labeling space while respecting the precomputed network layouts, such as those obtained by the conventional force-directed approach. Voronoi tessellation is widely employed for this purpose because it allows us to spare more space around the given samples according to their spatial density. Pulo [20] proposed the concept of recursive Voronoi tessellation to introduce more labels to the conventional forced-directed approach in the context of network visualization. Another application along these lines was developed by Wu et al. [21], who used weighted Voronoi tessellation for space decomposition, specifically for embedding annotation labels around schematized metro networks. Moreover, Brivio et al. [22] incorporated centroidal Voronoi tessellation to evenly distribute image sets within the limited screen space.

Indeed, centroidal Voronoi tessellation is useful for more aggressively sparing space around each network node in the force-directed layout. This is because centroidal Voronoi tessellation usually equalizes the areas of Voronoi cells by moving each sample to the center of the corresponding cell. Thus, this is more likely to rearrange sufficient space around each node and can avoid unnecessary overlap between the annotation labels. For example, Lyons et al. [23] iteratively applied additional forces originating from centroidal Voronoi tessellation to move the labels while keeping the network layout sufficiently close to the original one. Gansner and North [24] introduced a new force extracted from centroidal Voronoi tessellation and resolved the overlap between annotation labels as a postprocess. Nevertheless, such existing approaches usually compose the Voronoi tessellation only once during the network annotation and thus often limit the merit of the space-partitioning techniques insofar as they introduce redundant labeling space to resolve the associated label conflicts. Notably Didimo and Montecchiani [25] recently employed space partitioning based on conventional parallel-axis treemaps to explore network layouts. This could potentially be applied to our problem, although it does not provide enough degrees of freedom in the placement of node labels in the screen space.

Other labeling techniques, such as internal and external labeling techniques, are also relevant to our work. Internal labeling places annotation labels close to the target network nodes whereas external labeling attempts to place annotation labels in the predefined regions, usually boundary regions around the central content window. For practical applications, algorithms have also been developed specifically for annotating schematic networks such as railway and metro maps [26, 27, 28]. The conventional force-directed approach itself has also been extended to the map-labeling problem: mutual overlap between the labels was removed by iteratively applying repulsive forces inherited from the Coulomb model [29].



(a) (b)

Figure 2: Label placement of an artificial product co-purchasing network obtained by (a) the conventional force-directed algorithm and (b) centroidal Voronoi tessellation.

In this study, we explore a hybrid approach of force-directed algorithms and Voronoi-based space-partitioning techniques, which means that we pursue an adaptive compromise between the two techniques. Furthermore, hierarchical network structures are introduced for better control of the overall network layouts.

3. Overlap-Free Annotation of Networks

As described earlier, our approach can be considered a hybrid one in the sense that it examines a reasonable compromise between the network layouts obtained by the conventional force-directed algorithm and those by the centroidal Voronoi tessellation. In our implementation, we initialize the network layout using the force-directed algorithm and then gradually adjust the node positions accordingly, so that the layout remains similar to the simple force-directed one after introducing forces inherited from the centroidal Voronoi tessellation. This section first explains the force-directed formulation and centroidal Voronoi tessellation, which is followed by our proposed hybrid approach for seeking a reasonable compromise between the forces provided by the two layout approaches.

3.1. Force-directed algorithm

To establish an effective initial layout, we employ the conventional force-directed algorithm. Suppose that we have n network nodes $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1}$. We introduce the force-directed model, which simulates the physical dynamics of a coil spring to each edge. This is usually formulated using Hooke's law, as follows:

$$\boldsymbol{F}_d(\boldsymbol{v}_i, \boldsymbol{v}_j) = k_d(|\boldsymbol{v}_j - \boldsymbol{v}_i| - l_0)(\boldsymbol{v}_j - \boldsymbol{v}_i), \tag{1}$$

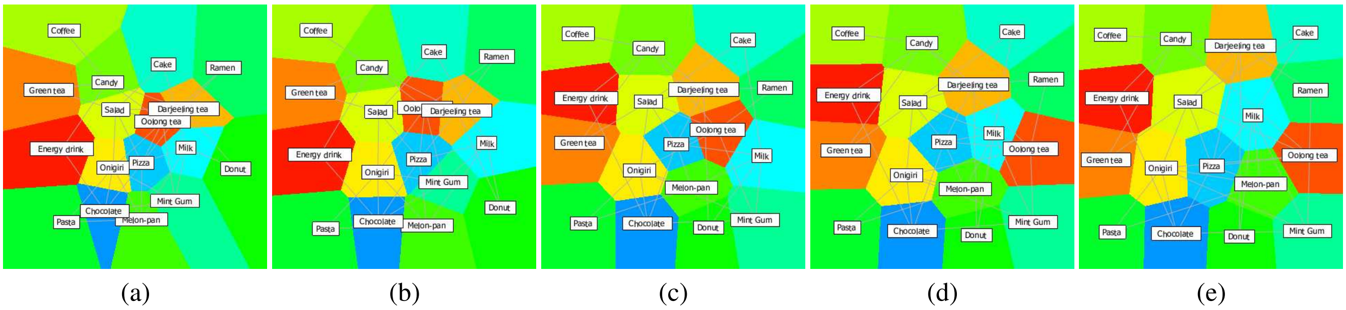


Figure 3: Transition from the force-directed layout to that of centroidal Voronoi tessellation. (a) $\alpha = 0.1$. (b) $\alpha = 0.3$. (c) $\alpha = 0.5$. (d) $\alpha = 0.7$. (e) $\alpha = 0.9$. Here, $(1 - \alpha)$ and α represents the weights of the force-directed algorithm and centroidal Voronoi tessellation.

where k_d represents the constant of the spring, l_0 is the original length of the spring, and $|\boldsymbol{v}_i - \boldsymbol{v}_j|$ is the Euclidean distance between the connected end nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. In our implementation, k_d is set to be 1.0 by default, and l_0 is set to be inversely proportional to the number of network nodes within the screen space. In this formulation, we apply the attractive force to all the pairs of nodes that are connected by edges.

Moreover, each node (i.e., $\boldsymbol{v}_i$) is assumed to have an electrical charge and takes a repulsive force exerted by another node (i.e., $\boldsymbol{v}_j$) according to the corresponding distances, which is defined as

$$\boldsymbol{F}_r(\boldsymbol{v}_i, \boldsymbol{v}_j) = -k_r(\boldsymbol{v}_j - \boldsymbol{v}_i)/|\boldsymbol{v}_j - \boldsymbol{v}_i|^2, \qquad (2)$$

where k_r corresponds to the electric force constant (i.e., Coulomb's constant) and set to be 0.1 by default in our setup. In summary, we compute the sum of these attractive and repulsive forces to each network node as

$$\boldsymbol{F}_s(\boldsymbol{v}_i) = \sum_{j \in N_i} \boldsymbol{F}_d(\boldsymbol{v}_i, \boldsymbol{v}_j) + \sum_{k \in V - \{i\}} \boldsymbol{F}_r(\boldsymbol{v}_i, \boldsymbol{v}_k), \qquad (3)$$

where N_i represents an index set of nodes adjacent to $\boldsymbol{v}_i$ and $V - \{i\}$ represents an index set of all the nodes, excluding i. We apply this force to each node until the network layout achieves an equilibrium state, where we can empirically generate a visually plausible result that avoids excessive visual clutter, such as edge crossings. Unfortunately, even with this force-directed approach, we cannot fully avoid overlaps between the nodes if they are associated with rectangular text labels, as shown in Figure 2(a). In our algorithm, we use this layout as an initial state for exploring optimized layouts of such an annotated network.

3.2. Centroidal Voronoi tessellation

Centroidal Voronoi tessellation allows us to effectively distribute seed points uniformly within a finite domain, and it can thus potentially mitigate unwanted overlaps between node labels. In centroidal Voronoi tessellation, we usually partition the space by repeatedly conducting three steps:

1. Compute the Voronoi tessellation with seed points
2. Identify the barycenter of each Voronoi cell
3. Move a seed point to its corresponding barycenter

These steps are repeated until we can finalize the uniform partitioning of the domain.

In Step 1), we first compute the Voronoi tessellation by referring to the coordinates of each node $\boldsymbol{v}_i$ as a seed point. To compose the Voronoi diagram, a hardware-assisted algorithm [30] is employed that accelerates the computation by placing 3D cones of different colors centered at the given seed points and projecting them onto the screen from the top. The barycenter of the Voronoi cell around $\boldsymbol{v}_i$ is computed as $\boldsymbol{g}(\boldsymbol{v}_i)$ in Step 2). In our implementation, the system scans the frame buffer to collect a set of pixels of the specific color for each Voronoi cell and computes the barycenter as the average of the corresponding pixel coordinates. In Step 3), each seed point $\boldsymbol{v}_i$ is moved to the barycenter of the corresponding Voronoi cell $\boldsymbol{g}(\boldsymbol{v}_i)$ to uniformly distribute them within the screen space. This amounts to applying the following forces to each node $\boldsymbol{v}_i$:

$$\boldsymbol{F}_v(\boldsymbol{v}_i) = C(\boldsymbol{g}(\boldsymbol{v}_i) - \boldsymbol{v}_i), \qquad (4)$$

where C is a predefined constant that controls the strength of this force and is set to be 10 by default in our implementation.

These three steps are repeated until each seed point (i.e., network node) reaches its equilibrium position. Figure 2(b) depicts an example, in which the network nodes, together with their labels, are displaced using centroidal Voronoi tessellation.

3.3. Hybrid approach

As shown in Figure 2(a), the conventional force-directed algorithm can successfully alleviate visual clutter while it cannot explicitly spare sufficient space around the network nodes. Thus, the associated node labels may overlap with each other, especially when the network is unexpectedly dense in its topological connectivity. On the other hand, although centroidal Voronoi tessellation provides us with an effective means of uniformly distributing the network nodes, it is more likely to align the nodes along the horizontal and vertical directions and cannot fully eliminate visual clutter including edge overlaps, as shown in Figure 2(b). Thus, our challenge here is to seek a plausible compromise between the two layouts. We are able to achieve this compromise by smoothly transforming the initial force-directed layout to that based on centroidal Voronoi tessellation [1].

We accomplish this goal by applying the weighted sum of the forces exerted by the two algorithms (cf. Eqs. (3) and (4)) to each node $\boldsymbol{v}_i$, as follows:

$$F_h(\boldsymbol{v}_i) = (1 - \alpha)F_s(\boldsymbol{v}_i) + \alpha F_v(\boldsymbol{v}_i), \qquad (5)$$

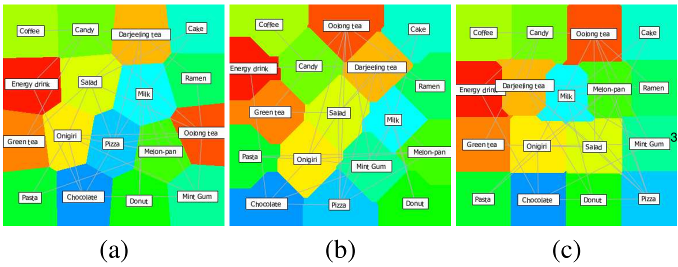


Figure 4: Network layouts based on centroidal Voronoi tessellation with the three distance metrics. (a) Euclidean distance. (b) Manhattan distance. (c) Chebyshev distance.

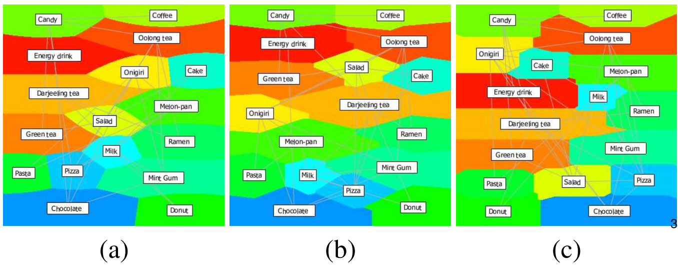


Figure 5: Network layouts based on centroidal Voronoi tessellation when the aspect ratios of text labels are considered individually in terms of nodes. (a) Euclidean distance. (b) Manhattan distance. (c) Chebyshev distance.

where α is the blending ratio between the two forces. Furthermore, to maximally retain the spatial layout obtained through the force-directed algorithm, we start with $\alpha = 0$ and then gradually increase it so that we can smoothly transform the initial layout to that of centroidal Voronoi tessellation. This is accomplished by computing the total sum of the forces in Eq. (5) every time we displace the network nodes and by updating the weights by a small amount δ once the sum becomes less than a predefined threshold. In our approach, δ is set to be 0.02 by default. This allows us to respect the original layout of the network while sparing more space around the nodes for their annotations.

Figure 3 shows how we can transform the initial network layout by incorporating the influence of centroidal Voronoi tessellation, in which the overlaps between the network nodes are gradually resolved during the transition while respecting the relative positions of the network nodes in the original layout. In our approach, we first increase α to 0.6, and further increase it up to 0.9 if we still have overlaps between node labels in the network. This also helps us avoid unwanted edge overlaps, which often arise from layouts based only on centroidal Voronoi tessellation, as shown in Figure 2(b).

4. Enhancements of Distance Metrics

In this section, we detail several enhancements in our choice of distance metrics. We made these enhancement to establish a better arrangement of annotation labels. We also describe additional enhancements, which were made to guarantee the overlap-free network labeling and to control the placement of labels through manual intervention.

4.1. Selection of the distance metric

As the text annotations, rectangular labels are commonly employed to provide concise explanations for the respective network nodes. Here, we have tested the Euclidean, Manhattan, and Chebyshev distances in order to identify the most appropriate distance metric when placing these rectangular node labels.

Suppose that we consider the distances between the two nodes $\boldsymbol{v}_i = (x_i, y_i)$ and $\boldsymbol{v}_j = (x_j, y_j)$. In fact, these three types of distances are defined as follows:

$$
\begin{aligned}
d_E(\boldsymbol{v}_i, \boldsymbol{v}_j) &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} && (6)\\
d_M(\boldsymbol{v}_i, \boldsymbol{v}_j) &= |x_1 - x_2| + |y_1 - y_2| && (7)\\
d_C(\boldsymbol{v}_i, \boldsymbol{v}_j) &= \max(|x_1 - x_2|, |y_1 - y_2|) && (8)
\end{aligned}
$$

Here, the distance metrics d_E, d_M, and d_C correspond to the Euclidean, Manhattan, and Chebyshev distance metrics, respectively. Figure 4 presents the comparison between screen space Voronoi tessellations based on these three distance metrics. Recall that the Voronoi tessellation based on the Euclidean distance can be composed by arranging 3D cones centered at the respective nodes and projecting them from the top [30], as shown in Figure 4(a). Intuitively, the tessellation based on the Manhattan distance can be generated by replacing the cones with 3D square pyramids, in which the sides of each basement square are aligned with diagonal directions on the screen space, as presented in Figure 4(b). By contrast, the Chebyshev distance transforms the Voronoi tessellation by placing 3D square pyramids, in which the sides of the basement are aligned with horizontal and vertical directions, as shown in Figure 4(c).

It is clear from Figure 4 that the Chebyshev distance metric is more likely to produce rectangular Voronoi cells, as opposed to the polygonal and diamond cells generated by the Euclidean and Manhattan distance metrics. In this sense, the Chebyshev distance metric is preferable, because the node labels best fit the shape of the corresponding Voronoi cells. However, the metric still often fails to completely enclose the rectangular labels, which will be discussed in the next section.

4.2. Distance anisotropy in sparing labeling space

As the network layout in Figure 4(c) shows, the text label “Darjeeling tea” exceeds the boundary of the allocated Voronoi cell. Because this label contains many letters, and thus its shape is horizontally longer than other labels. This observation leads us to the idea of adaptively changing the aspect ratios of such Voronoi cells according to the size of the corresponding text labels. In our approach, this is achieved by incorporating an appropriate anisotropic metric, which is a modified version of the Chebyshev distance metric:

$$
d'_C(\boldsymbol{v}_i, \boldsymbol{v}_j) = \max(|x_1 - x_2|, \alpha|y_1 - y_2|), \qquad (9)
$$

where α indicates the aspect ratio of the horizontal side of the corresponding label with respect to its vertical side.

Nonetheless, exactly computing the distance based on this metric is quite difficult if we locally change the aspect ratio by referring to the labels of the network nodes contained in

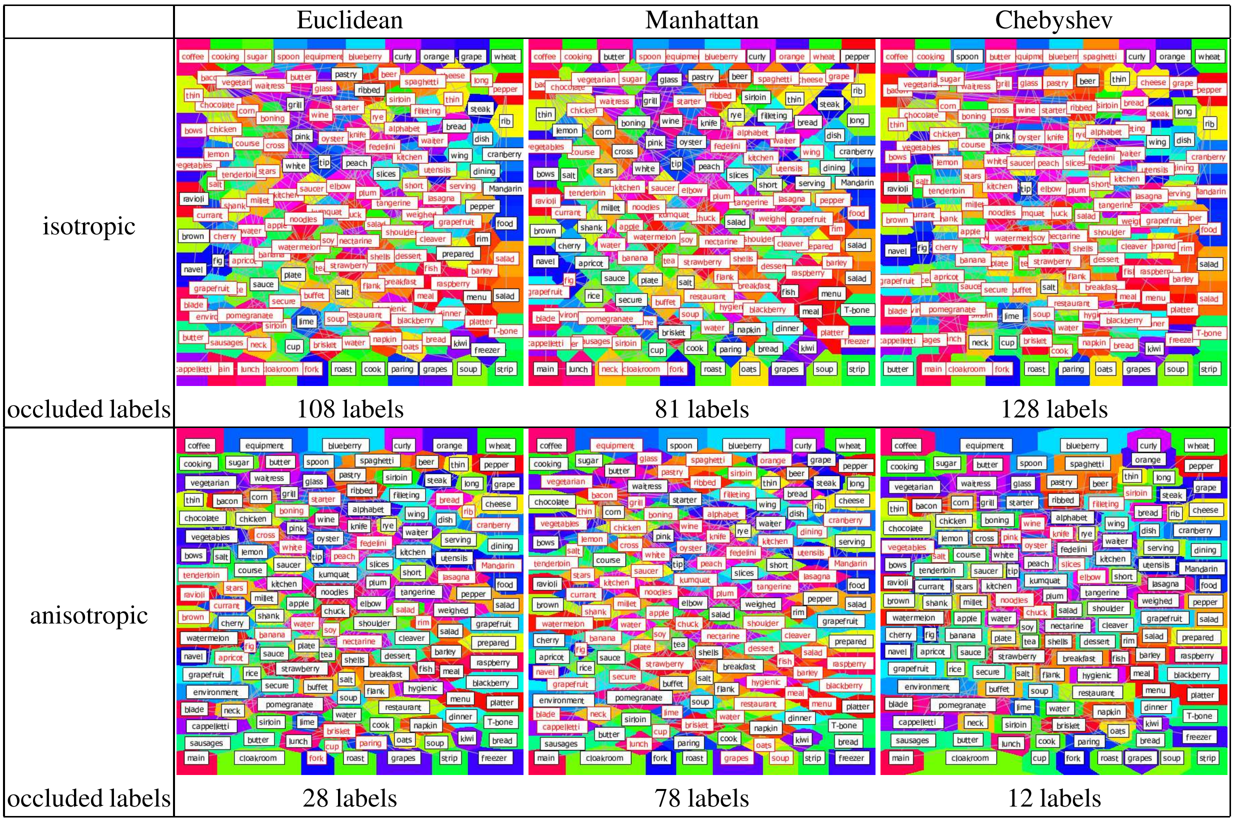


Figure 6: Label packing based on centroidal Voronoi tessellation. Labels colored in red have overlap with other labels. The choice of the distance metric and anisotropy has a significant impact on the number of occluded labels.

the local neighborhood. Instead, in our approach, we approximate this anisotropic Chebyshev distance metric by scaling the square basement of the 3D pyramid according to the aspect ratio of the corresponding node label when composing the Voronoi tessellation. This means that the basement of the 3D pyramid at each node will be horizontally extended to fully accommodate the annotation label within its Voronoi cell. This is possible because we can count the number of letters in each text label and compute the corresponding aspect ratio beforehand. Figure 5 shows the Voronoi tessellations using the anisotropic versions of the three distance metrics. In particular, the Voronoi cells in Figure 5(c) respect the aspect ratios of the corresponding text labels and are best fit to their shapes among the three metrics.

We also tested the three distance metrics and their anisotropic versions by investigating how many labels they can accommodate without overlaps within the fixed screen space, as shown in Figure 6. The comparison indicates that the anisotropic version of the Chebyshev distance metric is the best in the sense that the associated Voronoi tessellation successfully accommodate the largest number of labels without unwanted occlusions. We therefore decided to employ this distance metric when composing the Voronoi tessellations of the screen space for annotation purposes.

4.3. Computing overlaps among the labels

In order to fully guarantee overlap-free label annotation over the networks, we compute the sum of overlaps between the labels during the transformation of the network layout. Suppose that the two labels at $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ have a rectangular overlapping area, where w_{ij} and h_{ij} represent the horizontal and vertical sides of that area, respectively. In this setup, we can compute the two sides of the label overlap as

$$w_{ij} = \max\{0, \min\{w_i/2 + w_j/2 - |x_{ij}|, w_i, w_j\}\}, \quad (10)$$

$$h_{ij} = \max\{0, \min\{h_i/2 + h_j/2 - |y_{ij}|, h_i, h_j\}\}, \quad (11)$$

where w_i and h_i represent the width and height of the label at $\boldsymbol{v}_i$, respectively, and w_j and h_j are those of $\boldsymbol{v}_j$. In addition, $x_{ij} = |x_i - x_j|$ and $y_{ij} = |y_i - y_j|$. Now, we can sum up $w_{ij} \cdot h_{ij}$ for all the pairs of the nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ to compute the total area of the label overlaps. If the sum of the overlaps vanishes, we can stop the iterative transformation of the network layout, because at that point we can fully guarantee the occlusion-free label annotation.

4.4. Selective annotation of network nodes

We also equipped our prototype system with an interface for interactively selecting network nodes to be annotated even when the layout transformation is still in progress. Users can select a single node by clicking it with a mouse or a group of nodes by enclosing a rectangular region with a box interface. This often facilitates attempts to manually control the placement of specific nodes, and especially clusters when annotating clustered networks. We also allow users to arbitrarily zoom in/out and translate the network in order to facilitate fine tuning of the network annotation on a limited screen space.

5. Adaptively Blending Network Layouts

The formulation we have developed so far is designed to interpolate between the two layouts obtained by the conventional force-directed approach and centroidal Voronoi tessellation, by employing a single ratio for blending the entire layout at all the network nodes [1]. However, this often incurs redundant space for labeling, especially around sparsely distributed nodes, as shown in Figure 7(b). This implies that inappropriate design of the blending ratios often degrades the quality of the network layout because the initial aesthetic layout cannot be plausibly retained. In this section, we detail how we transform the blending ratios individually at the network nodes, which leads to a better equilibrium state between the incorporated force-directed algorithms and centroidal Voronoi tessellation.

5.1. Laplacian smoothing for adjusting blending ratios

The problem of uniformly blending the aforementioned two forces is to explore the global compromise between the force-directed layout and that obtained from centroidal Voronoi tessellation with only the single blending parameter. This is explained by the fact that the density of the network nodes spatially changes within the screen space and thus the demand for additional labeling space differs accordingly. Empirically, it is better to find a local compromise that adaptively introduces the additional space around each node according to the demand. This leads us to the idea of propagating such space allocation request from one node to its neighbors by preparing the blending parameter for each node.

Let us summarize a naïve scheme for spreading the change in the blending ratios as follows. First, we compute the initial network layout with only the conventional force-directed approach. We then adaptively adjust the blending ratio of each network node to find an acceptable local equilibrium state between the two layouts. For example, let us denote the blending ratio of $\boldsymbol{v}_i$ by α_i. We can implement adaptive blending of the two layouts by increasing α_i by a small amount (0.02 by default in our implementation), if the annotation label of $\boldsymbol{v}_i$ still needs more space for its placement. Furthermore, we smooth out α_i with the blending parameters of its neighbors so that we can propagate the space allocation request to the nodes in its vicinity. In our implementation, we initialize the blending ratio α_i to 0.0 and iteratively update it through the local smoothing process until the label overlaps are fully resolved.

As for the smoothing process, we apply the typical Laplacian smoothing approach at this stage to find the smooth transition of the blending ratios. This is achieved by updating α_i to α'_i using the following equation:

$$\alpha'_i = \sum_{j \in N_i} \alpha_j / |N_i|, \tag{12}$$

where N_i represents an index set of the neighbor nodes around $\boldsymbol{v}_i$ and $|N_i|$ indicates its size, respectively. This amounts to replacing the previous blending ratio at $\boldsymbol{v}_i$ with the average ratios of its neighbors. The results obtained by the Laplacian smoothing operation are shown in Figure 7(c).

5.2. Two-step smoothing for adjusting blending ratios

In practice, Laplacian smoothing successfully transforms an initial network layout obtained by the conventional force-directed approach (Figure 8(a)) to a visually acceptable one (Figure 8(b)). However, the resultant network still contains redundant space around the network nodes, especially when they are sparsely arranged in the screen space. This is because the blending ratios will be excessively smoothed out when we apply Laplacian smoothing many times.

In order to preserve the original distribution of the blending ratios at the respective nodes to a certain extent while properly smoothing them out, we employed Taubin's two-step smoothing [31, 32] instead. The advantage of this smoothing operation is that it works as an approximate low-pass filter and thus never incurs unwanted shrinkage problems, as the conventional Gaussian filters do, especially when we think of the spatial distribution of the local blending ratios as a signal defined over the screen space. Taubin's smoothing consists of alternating shrinking and expanding steps, which can be formulated as follows:

$$\begin{aligned} \alpha'_i &= \alpha_i + \lambda\{(\sum_{j \in N_i} \alpha_j / |N_i|) - \alpha_i\} \quad \text{and} \\ \alpha''_i &= \alpha'_i + \mu\{(\sum_{j \in N'_i} \alpha'_j / |N'_i|) - \alpha'_i\}, \end{aligned} \tag{13}$$

where $0 < \lambda < -\mu$. The first equation corresponds to the shrinking step, which updates α_i to α'_i, and the second equation represents the expanding step, which replaces α'_i with α''_i. Note that $|N_i|$ and $|N'_i|$ represent the sizes of the neighbors around $\boldsymbol{v}_i$ at each step. See [31] for possible choices of λ and μ. We employed the parameter setup proposed in [32] among the possible choices. By applying this two-step smoothing operation, we can obtain a network layout that is more compact and closer to the original force-directed layout, as shown in Figure 8(c).

5.3. Selecting nodes in the neighborhood

The last technical issue we have to consider is how to select the neighbor nodes in the smoothing process described above. In our approach, we prefer to collect spatially close neighbors instead of topologically adjacent ones, because node labels that are closer to the current node are more likely to collide with that of the current node. A natural solution is to collect neighbors within a specific radius around the current node. Although we tested this solution, the number of neighbors considerably varies according to spatial density, and it can become zero in cases where the current node is considerably isolated from others.

Thus, we select the k-nearest neighbors at each network node, as shown in Figure 9. Throughout this paper, we empirically set $k = 8$, because this setup produces relatively stable results. As for the distance metric, we again employ the anisotropic version of the Chebyshev distance as our distance metric.

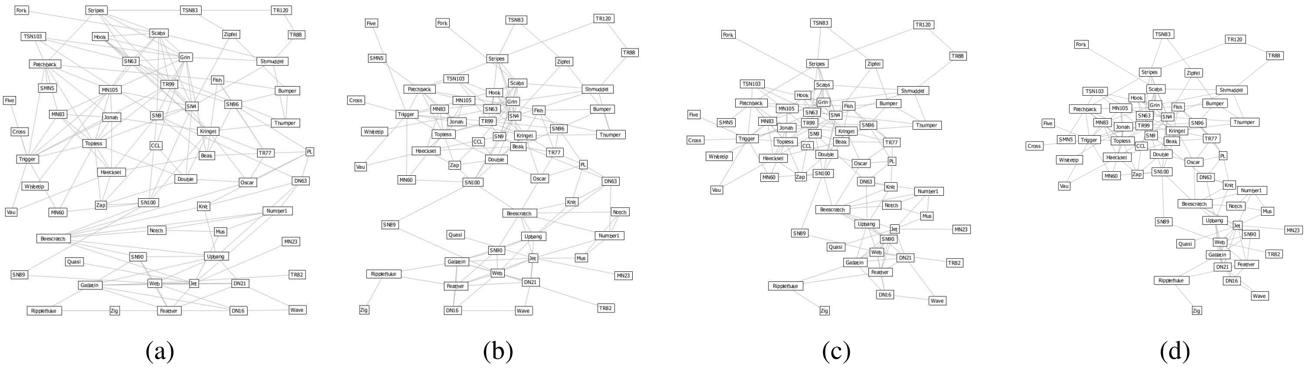


Figure 7: Comparison between the layouts of annotated social networks of dolphins [3]. (a) Layout obtained by centroidal Voronoi tessellation. (b) Transforming a single ratio to blend the entire layout. Transforming individual ratios individually at the network nodes to adaptively blend the layout using (c) the Laplacian filter and (d) the two-step filter. Note that the original force-directed layout is shown in Figure 1(a).

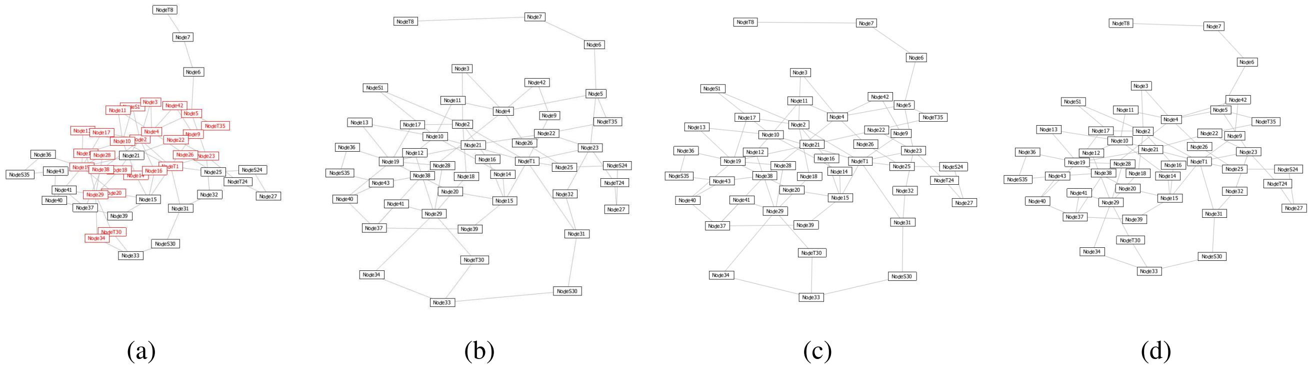


Figure 8: Difference between network layouts obtained by (a) the conventional force-directed approach, (b) Laplacian smoothing with the Euclidean distance metric, (c) Taubin's two-step smoothing with the Euclidean distance metric, and (d) Taubin's two-step smoothing with the Chebyshev distance metric.

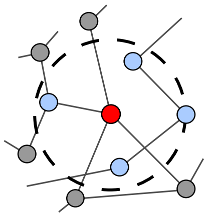


Figure 9: Definition of the k-neighbors around the the node.

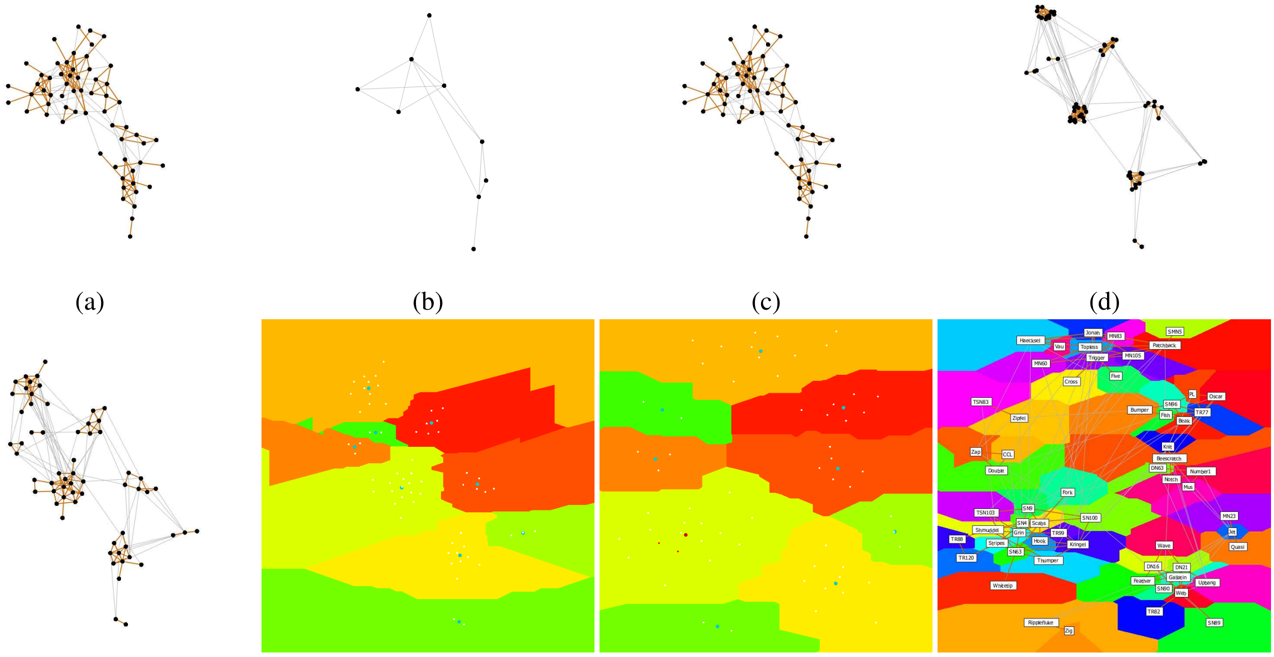
Figures 8(c) and (d) elucidate the difference between the Euclidean and Chebyshev distance metrics in this context. The comparison shows that the Chebyshev distance looks better here because it can successfully spare the horizontally elongated labeling space around the network nodes and keep the entire network layout more compact in the end. Figures 7(c) and (d) show another comparison between one-step Laplacian smoothing and Taubin's two-step smoothing with the Chebyshev distance used for exploring the k-neighbors. In this case, we can visually confirm that the two-step smoothing maximally retains the original layout produced by the force-directed approach.

6. Extension to Clustered Networks

This section presents the extension of the aforementioned node-labeling approach to clustered networks. It is now common to extract meaningful small communities in the network as a preprocess by evaluating the significance of the network nodes and edges, especially when the network is relatively large in size and its layered structure is unlikely to be clear from the beginning. This suggests that it is important to annotate this type of clustered network while maximally retaining its overall layouts, although doing so becomes more difficult primarily because we have to fit the entire network within a limited screen space. In our implementation, we first compose the aggregated version of the network by contracting the nodes in each cluster into a single node. We then distribute the network nodes in each cluster in the neighborhood of the corresponding contracted node. Finally, we spare the labeling space around the individual nodes while distributing the nodes of the aggregated network uniformly in the screen space.

6.1. Contracting clusters of network nodes

First, we perform a cluster analysis of the network to represent it in a hierarchical fashion. For this purpose, we need to evaluate the contribution of the respective network primitives such as nodes and edges and then extract a set of meaningful



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

(e) (f) (g) (h)

Figure 10: Labeling the clustered network representing social relationships between dolphins [3]. (a) A clustered network layout based on edge betweenness centrality. (b) An aggregated network where each node corresponds to a cluster of nodes. (c) The force-directed layout of the aggregated network. (d) Nodes in each cluster are randomly arranged within the neighborhood of the corresponding contracted node. (e) Nodes in each cluster are rearranged using the force-directed approach while keeping their barycenter identical with the corresponding contracted node. (f) Voronoi cell decomposition associated with the aggregated network. (g) Rearranged cell decomposition after hierarchically applying adaptive blending to both the original and aggregated network. (h) Finalized layout of the annotated clustered network.

clusters by referring to each primitive's level of contribution. As for the amount of contribution, we employ the edge betweenness centrality, which is defined as the degree to which the edge serves as a part of every shortest path over the network. We compute the edge betweenness centrality of the given network using algorithms developed by Brandes [33] and cut out the edges if they have larger betweenness centrality weights than some threshold. In our implementation, the corresponding threshold is carefully selected using a binary search of the sorted list of edge betweenness centrality weights of the network, in such a way that the individual decomposed clusters contain approximately 10 nodes on average. Figure 10(a) shows an example in which the edges colored in orange stay connected after the network decomposition based on the edge betweenness centrality, which means that the orange edges have a lower centrality value than the selected threshold.

After decomposing the entire network into clusters, we compute the barycenter of each cluster as its representative position and contract the cluster to a single node, as shown in Figure 10(b). Simultaneously, we connect a pair of such contracted nodes with a link if the clusters in the corresponding pair share at least one edge in the original network. This contraction step allows us to delineate the backbone structure inherent in the given network through the network clustering process. The aggregated version of the network will be used as a starting point for spatially rearranging the given network while maximally respecting its clustered structure, which was obtained through the edge betweenness centrality computation. Note that in this clustering process, we can employ different edge weights, including other types of centrality measures, according to the requirements and preferences associated with the visual analysis of the network data.

6.2. *Hierarchical layout of clustered networks*

Our next step is to rearrange the aggregated version of the network so that we can properly arrange the entire network within the screen space. In our implementation, we simply apply the conventional force-directed approach to the aggregated network, as shown in Figure 10(c), while also permitting users to optionally relocate the contracted nodes as they like in our prototype system. After having improved the aggregated network layout, we restore the connectivity of the original network by referring to the aggregated version. This is accomplished by replacing each contracted node with original network nodes in the corresponding cluster and randomly distributing them within a small neighborhood of that contracted node, as shown in Figure 10(d).

To further improve the layout of the clustered network, we again apply the force-directed approach but we apply it individually to each cluster of nodes, as shown in Figure 10(e). This hierarchical layout is achieved by first applying the conventional repulsive and attractive forces to the nodes in the respective clusters and then transforming them under the constraint in such a way that their barycenter always coincides with the position of the corresponding contracted nodes in the aggregated network. Note that this constraint will be retained in the remainder of this hierarchical layout of the network. In this process, we do not take into account repulsive forces between the nodes if they belong to different clusters. Moreover, the original length of the spring is again set to be inversely proportional to the number of network nodes within the screen space, and the length thus becomes smaller when we replace the aggregated network with the original one.

6.3. *Labeling nodes of clustered networks*

At this point, we are ready to annotate the nodes of the clustered network while preserving its overall layout and avoiding mutual overlap between the labels. This is again implemented by introducing centroidal Voronoi tessellation as the space-partitioning technique; however, this time, it is introduced to both the original and aggregated networks. Figure 10(f) shows the starting state of the Voronoi tessellation associated with the aggregated network layout, in which each site that contains one contracted node together with the original nodes in the corresponding cluster is drawn in a different color. In our implementation, we apply the aforementioned smoothing approach to the individual nodes of the original network. By contrast, we simply just increment the blending ratios of the contracted nodes in the aggregated network on a step-by-step basis when the labels of its corresponding member nodes have overlap with others. This is because the number of contracted nodes in the aggregated network is originally small and smoothing their blending ratios is of no use. It should be remembered that we also maintain the constraint imposed on the nodes in each cluster by applying the same amount of displacement to them every time we rearrange the nodes in the two networks. Figure 10(g) illustrates the final state of the space partitioning associated with the aggregated network, and Figure 10(h) presents the corresponding layout of the annotated network. We can confirm from the results that the final layout of the annotated network maintains the clustered structure of the original layout while successfully avoiding mutual overlap between the annotation labels.

7. Experimental Results

This section presents examples of annotated network layouts and compares between networks of four typical topological types, followed by a discussion of the potential limitations of this approach. Our prototype system has been implemented on a desktop PC with 3.5 GHz 6-Core Intel Xeon E5 CPU and 32GB RAM, and its source code was written in C++ using boost graph library (BGL) for handling network data, OpenGL for graphics rendering, and Qt for system interface. We have equipped our system with an interface for selecting a subset of nodes in order to annotate them and for dragging the nodes to specific positions. Furthermore, additional forces applied to nodes around the boundary of the screen space in such a way that they can stay within that space if they are annotated with text labels. Datasets available at GraphViz (http://www.graphviz.org/) have been incorporated in our experiments.

7.1. *Network layout examples*

Figure 11 shows how our approach can improve the readability of the annotated network by adaptively blending the forces exerted by the force-directed approach and centroidal Voronoi tessellation. Figure 11(a) represents a conventional force-directed layout of the “b124” network, and Figure 11(d) corresponds to that obtained by centroidal Voronoi tessellation. We cannot avoid unwanted overlap between the text labels as shown in Figure 11(a). In Figure 11(d), by contrast, the text labels are more likely to be horizontally or vertically aligned, which causes distracted edge overlaps in the visualization of annotated networks. Figures 11(b) and (c) are the intermediate layouts, in which the single unique ratio is employed to blend the two layouts in Figure 11(c), and blending ratios are individually assigned to all the nodes in order to adaptively interpolate between the two layouts in Figure 11(b). The results indicate that the layout in Figure 11(b) is more aesthetic because it can locally retain the original force-directed layout around the nodes that still have sufficient labeling space. Unfortunately, however, the layout in Figure 11(c) becomes similar to that in Figure 11(d) if it cannot find sufficient labeling space around the nodes, and thus degrades the readability of the network data. Note that in the caption for Figure 11, T and O indicate the computation times (in seconds) and percentage of overlap with respect to the total area of labels, respectively. We use the same notation throughout the remainder of this paper.

Figure 12 demonstrates how we can arrange the clustered version of the annotated network “b106.” In this scenario, as described earlier in Section 6.1, we first decompose the entire network into several clusters by referring to the betweenness centrality value of each network edge (Figure 12(a)). We then compose the aggregated network by contracting the nodes and edges contained in each cluster into a single node (Figure 12(b)). After obtaining the force-directed layout of the aggregated network, we resolve each cluster into its component nodes while clarifying the characteristic structure of the original network (Figure 12(c)). Even in this case, we can successfully incorporate the space-partitioning technique to completely eliminate the label overlaps while maximally preserving the existing hierarchical network layout (Figure 12(d)).

Figure 13 provides a comparison with the network layout calculated using a previous approach. The algorithm developed by Gansner et al. [18] makes it possible to transform the conventional force-directed layout presented in Figure 13(a) to that presented in Figure 13(b). In fact, their algorithm can successfully eliminate the mutual overlap between the annotation labels by inserting additional space around the nodes. Nonetheless, the size of the newly introduced space often becomes redundant because it strictly preserves the relative positioning of network nodes and edges and thus cannot maximize the space coverage of the screen space. Compared with this previous method, our approach plausibly seeks a better compromise between the layout obtained by the force-directed approach (Fig-

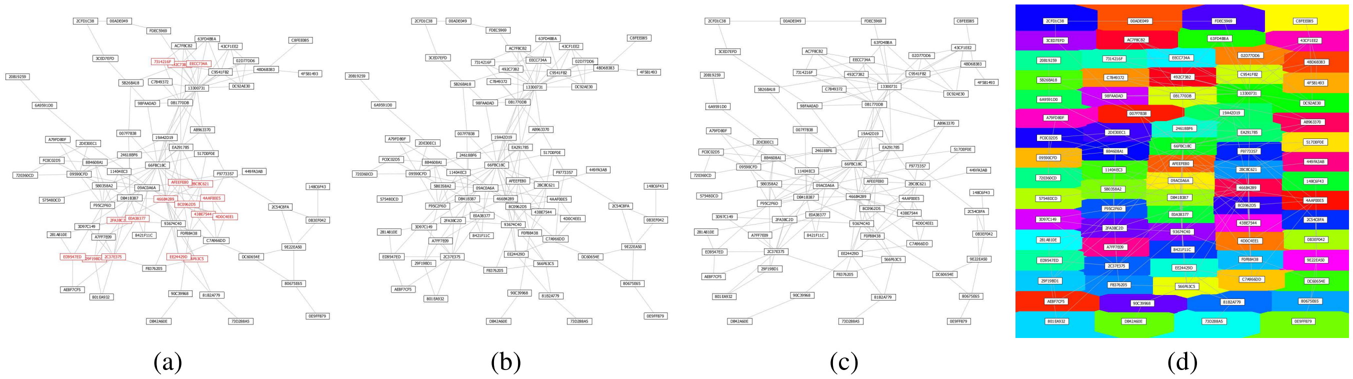


Figure 11: Visualizing the "b124" network data ($|V| = 79$, $|E| = 281$, 1,024 × 1,024 pixels). (a) A layout obtained using the force-directed algorithm (T = 39.64 s, O = 1.20%). (b) A layout obtained by adaptively blending the force-directed layout and centroidal Voronoi layout individually with respect to the network nodes. (T = 1.35 s, O = 0.00%). (Our approach) (c) A layout obtained by uniformly blending the force-directed layout and centroidal Voronoi layout over the network. (T = 14.51 s, O = 0.00%). (d) A layout obtained using centroidal Voronoi tessellation (T = 33.90 s, O = 0.00%). Note that the layouts in (b), (c), and (d) are transformed from that in (a).

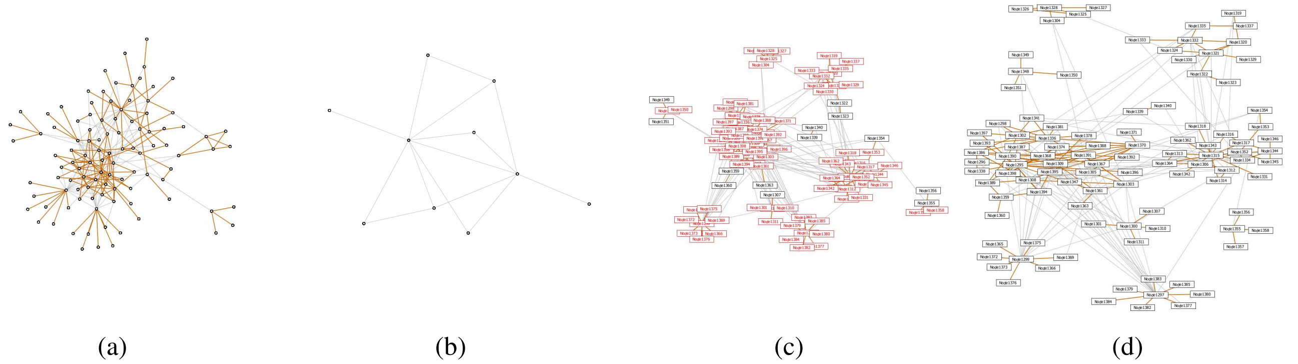


Figure 12: Visualizing the "b106" clustered network data ($|V| = 104$, $|E| = 230$, 1,024 × 1,024 pixels). (a) Clustering the network using edge betweenness centrality. Orange edges are those contained in the individual clusters. (b) Force-directed layout of the aggregated network, in which the nodes correspond to the respective clusters. (c) Hierarchical representation of the force-directed clustered network (O = 14.74%). (d) Finalized layout of the clustered network obtained through centroidal Voronoi tessellation (T = 26.80 s, O = 0.00%).

ure 13(a)) and that obtained by centroidal Voronoi tessellation (Figure 13(d)), as shown in Figure 13(c).

Our system also provides us with a means of interactively turning on/off the text labels by selecting the corresponding network nodes. Accordingly, Figure 14(a) shows a case in which we are about to select a subset of nodes for annotation using rectangular selection. At the same time, we conducted a structure analysis of this network by computing the betweenness centrality value for each edge and decomposed the entire network into several clusters, as shown in Figure 14(b). After clarifying its hierarchical structure by applying the force-directed approach to this clustered network, as shown in Figure 14(c), our technique can successfully spare labeling space around the selected subset of nodes by fully taking advantage of the screen space, as illustrated in Figure 14(d).

7.2. *Comparison between four typical types of networks*

To evaluate the feasibility of our approach, we tested four typical network types; banana trees, random networks, scale-free networks, and small world networks. We also applied our approach to both nonclustered and clustered networks to expose the difference in their layouts according to the type of network representation. Figure 15 presents a comparison between the layouts of four nonclustered annotated networks. Our approach can establish a reasonable compromise between the layout obtained through the force-directed approach and that obtained through centroidal Voronoi tessellation, and it can thus successfully maintain the overall configuration of force-directed layouts while sparing sufficient labeling space around the respective nodes. As for the comparison between the four types of networks, our approach can illuminate the concentric structure inherent in the banana trees even when they are not hierarchically rearranged. However, it cannot fully reflect the underlying structures in the other three types of networks.

For the clustered networks, our approach provides an additional means of controlling the overall network layout. Figure 16 presents a comparison in which clustered networks of the four typical types are annotated with text labels. In this case, we computed the betweenness centrality of the network edges first and then decomposed the networks into several clusters as a preprocess, as shown in the first row of Figure 16. This network analysis helps us compose aggregated versions of the networks, as exhibited in the second row. After this step, we recovered the original network topology while respecting

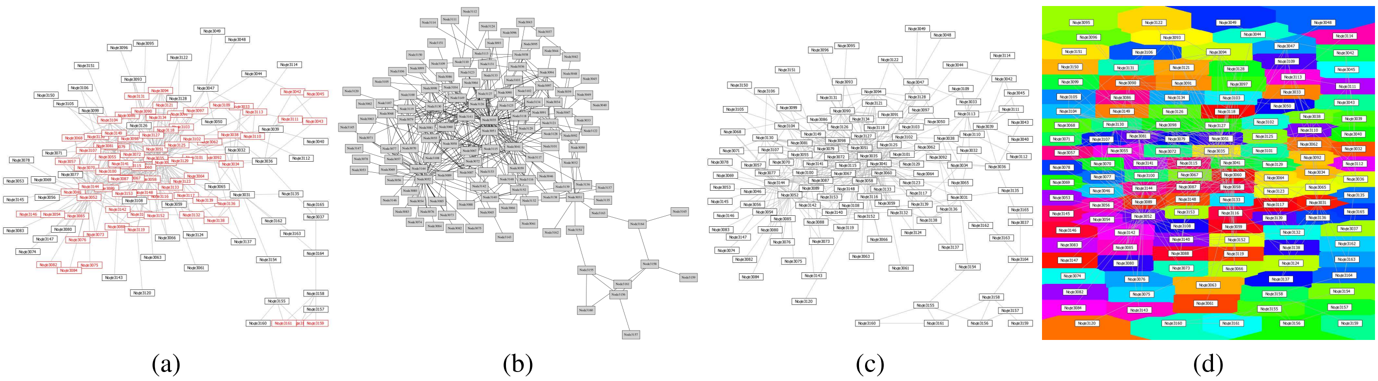


Figure 13: Visualizing the "b143" network data ($|V| = 135, |E| = 366$, 1,024 × 1,024 pixels). (a) A layout obtained using the force-directed algorithm. (b) A layout obtained using the algorithm proposed by Gansner et al. [18]. (c) A layout obtained using our approach, in which we adaptively blend the force-directed layout and that obtained by centroidal Voronoi tessellation (T = 37.87 s, O = 0.00%). (d) A layout obtained using centroidal Voronoi tessellation only.

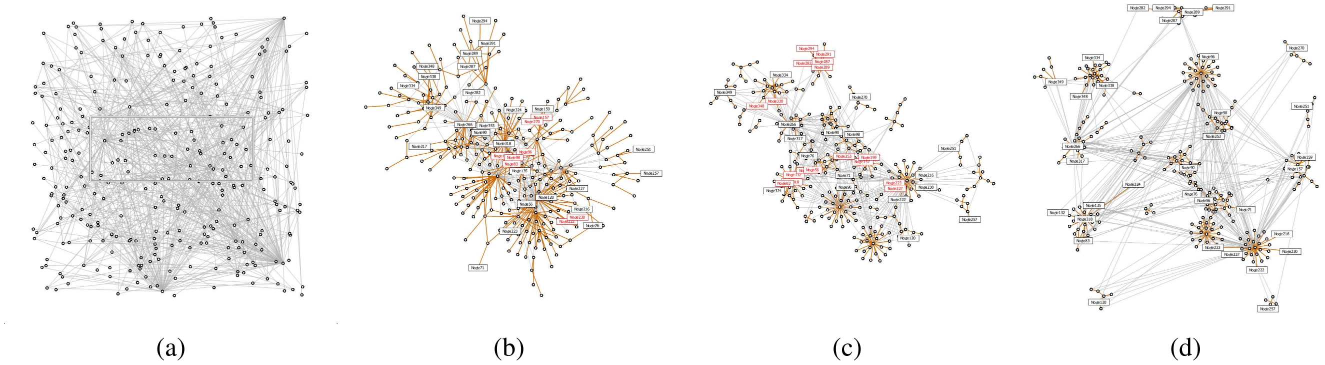


Figure 14: Interactively annotating the "b102" network data ($|V| = 302, |E| = 611$, 1,024 × 1,024 pixels). (a) Activating text annotations by selecting a subset of nodes using rectangular selection. (b) Cluster analysis of the network while the selected nodes are annotated with text labels. (c) Hierarchical layout of this partially annotated clustered network. (d) Finalized layout of annotations associated with the selected subset of nodes (T = 83.57 s, O = 0.00%).

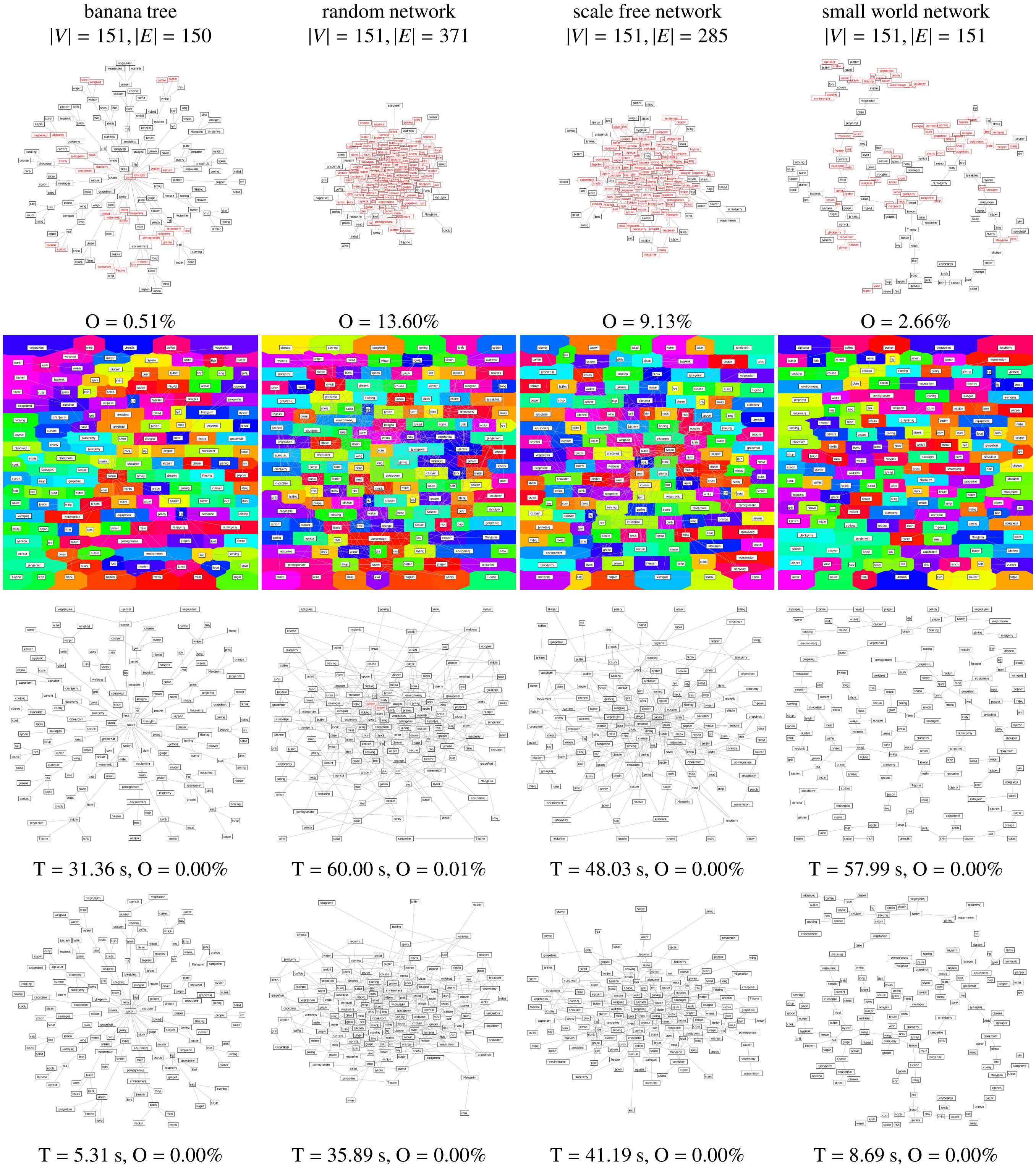
the layouts of the aforementioned aggregated networks, as presented in the third row, and augmented labeling space around the nodes until all the labels were placed without distracting overlaps, as shown in the fourth row. In this case, we first delineated the backbone structure of the respective networks as aggregated forms, and as a result the placement of text labels is more likely to reflect the underlying structure hidden behind the networks. For example, we can explicitly extract concentric subnetworks from banana trees, hub nodes from scale-free networks, and sparse backbone networks from small world networks, as illustrated in Figure 16. The placement of node labels provides a better outline of such backbone structures of the respective networks. Furthermore, the aggregated representation provides us with more flexibility in properly arranging entire networks within a limited screen space.

7.3. Discussion

Of course, the number of labels that can be placed basically depends on the availability of the screen space. However, placing an excessive number of annotation labels usually results in network layouts that are quite close to those obtained directly by centroidal Voronoi tessellation. This inevitably degrades the quality of network layouts, because edges are more likely to overlap with each other. Furthermore, this covers the majority of the screen space and thus hides the connectivity among the network nodes behind the text labels, which again lowers the readability of the annotated network layouts. Inferring a reasonable number of labels while establishing a better compromise between the amount of annotation and network readability is an important issue.

It is cumbersome to better control the ratios for blending the multiple network layouts, especially around the boundary of the screen space, because we adaptively change the ratio at each node by referring to those of the nodes in its local neighborhood. In practice, Voronoi cells around the boundary sometimes take up relatively large amounts of space, primarily because the changes in the blending ratios are not fully propagated to the rest of the network nodes. In this case, we can replace the two-step Taubin's smoothing with the single-step Laplacian smoothing in order to distribute such changes more broadly. However, this replacement will result in another problem: we cannot fully retain the original force-directed layouts. A more sophisticated scheme for updating blending ratios, possibly based on the hybrid formulation combining the Laplacian and two-step smoothing, may solve this technical problem.

Aggressively allowing more systematic user intervention is one of the promising solutions for editing annotated network layouts. For example, users can freely specify the range of



banana tree $|V| = 151, |E| = 150$

random network $|V| = 151, |E| = 371$

scale free network $|V| = 151, |E| = 285$

small world network $|V| = 151, |E| = 151$

O = 0.51%

O = 13.60%

O = 9.13%

O = 2.66%

T = 31.36 s, O = 0.00%

T = 60.00 s, O = 0.01%

T = 48.03 s, O = 0.00%

T = 57.99 s, O = 0.00%

T = 5.31 s, O = 0.00%

T = 35.89 s, O = 0.00%

T = 41.19 s, O = 0.00%

T = 8.69 s, O = 0.00%

Figure 15: Comparison between four typical types of nonclustered networks (1,024 × 1,024 pixels). First row: Force-directed layouts. Second row: Layouts obtained by centroidal Voronoi tessellation. Third row: Layouts obtained by blending two layouts uniformly with respect to the nodes. Fourth row: Layouts obtained by blending two layouts adaptively with respect to the nodes.

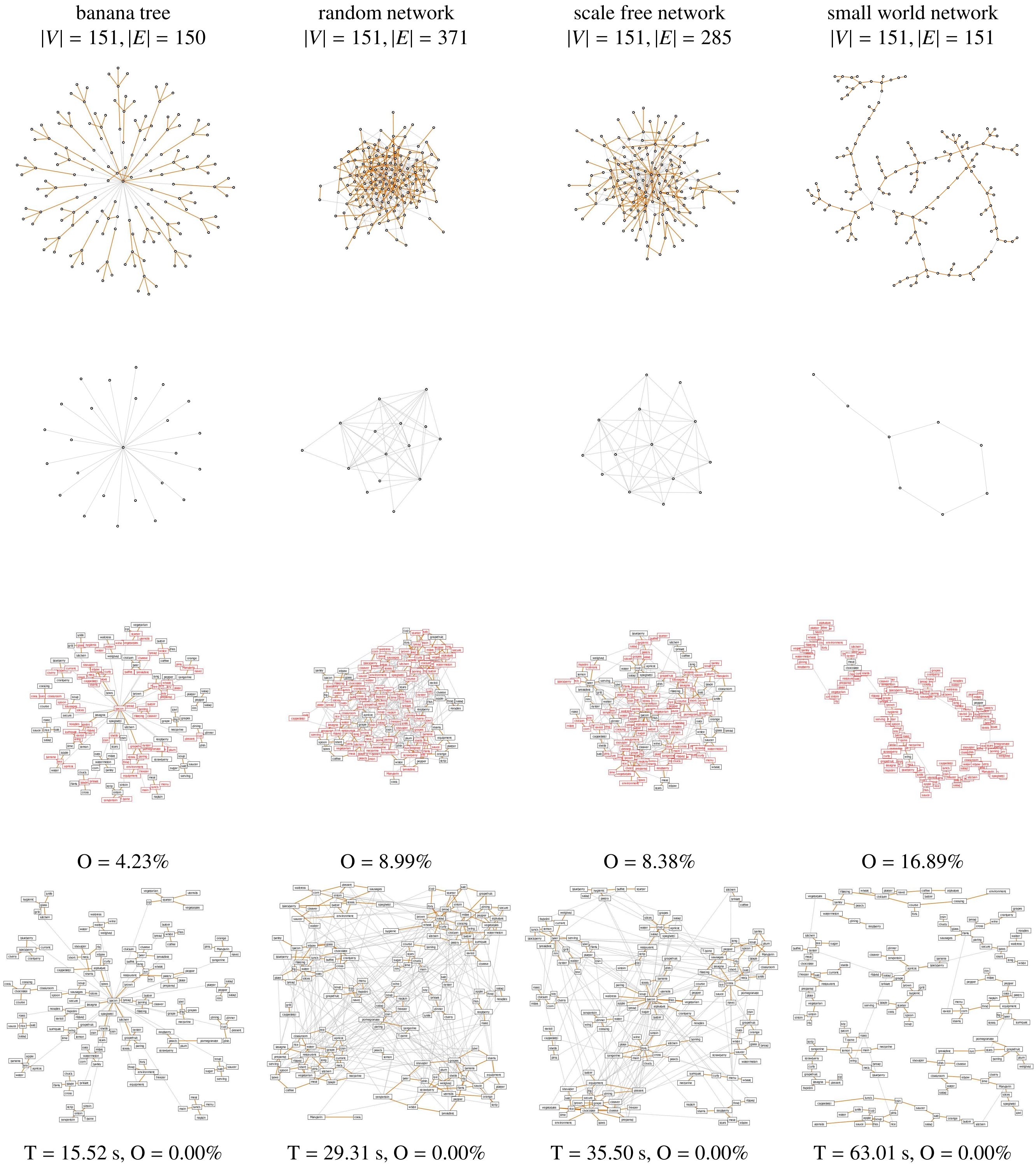


Figure 16: Comparison between four typical types of clustered networks. First row: Force-directed clustered layouts. Second row: Aggregated network layouts obtained by contracting the clusters into nodes. Third row: Hierarchical layouts of annotated clustered networks. Fourth row: Finalized layouts of annotated clustered networks. Note that the aggregated network layout of the small world network has been manually designed with the interface of our system.

labeling space within the screen space to properly control the placement of text labels. Asking users to select subsets of important nodes for annotation is a practical approach, especially for handling networks with a large number of nodes. Automatically selecting of such important nodes by referring to the network topology will be also helpful. Accelerating the overall computation, probably with the help of GPUs, will be in high demand in interactive environments as well.

8. Conclusion

In this paper, we have presented an approach for properly annotating network nodes by exploring a compromise between the layouts obtained by the conventional force-directed approach and centroidal Voronoi tessellation. The primary advantage of our technique is its capacity to effectively eliminate unacceptable overlaps among annotation labels while maximally retaining the initial layout of the network generated by the conventional force-directed approach. This can be accomplished by adaptively adjusting the ratio at each node in order to blend the aforementioned two forces according to the availability of its surrounding labeling space first, and then smoothing it out with those of its spatial neighbors through the shrinkage-free filter. We have further advanced our approach by extending our target from simple networks to clustered ones, so that we can explicitly visualize their inferred hierarchical structures.

Future research could investigate more sophisticated control over the blending ratios at the respective network nodes in order to further accelerate the annotation of large-scale networks. Incorporating more manual intervention in order to systematically guide annotated network layouts could also be an interesting subject for future research.

Acknowledgments

This study has been partially supported by the MEXT KAKENHI Grant Number 25120014, and JSPS KAKENHI Grant Numbers 16H02825, 17K12691, 26730061, and 15K12032. The project leading to this application has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 747985.

References

[1] R. Ishida, S. Takahashi, H.-Y. Wu, Interactively uncluttering node overlaps for network visualization, in: Proceedings of the 19th International Conference on Information Visualization (IV2015), 2015, pp. 200–205.

[2] R. Ishida, S. Takahashi, H.-Y. Wu, Adaptive blending of multiple network layouts for overlap-free labeling, in: Proceedings of the 20th International Conference on Information Visualization (IV2016), 2016, pp. 15–20.

[3] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, S. M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54 (2003) 396–405.

[4] J. Heer, D. Boyd, Vizster: Visualizing online social networks, in: Proceedings of IEEE Symposium on Information Visualization 2005 (InfoVis 2015), 2005, pp. 32–39.

[5] P. Eades, A heuristic for graph drawing, in: Proceedings of the Congressus Numerantium, Vol. 42, 1984, pp. 149–160.

[6] T. M. J. Fruchterman, E. M. Reingold, Graph drawing by force-directed placement, Software - Practice and Experience 21 (11) (1991) 1129–1164.

[7] G. D. Battista, P. Eades, R. Tamassia, I. G. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice Hall, 1998.

[8] Y. Hu, Efficient and high quality force-directed graph drawing, The Mathematica Journal 10 (2005) 37–71.

[9] P. Simonetto, D. Archambault, D. Auber, R. Bourqui, ImPrEd: An improved force-directed algorithm that prevents nodes from crossing edges, Computer Graphics Forum 30 (3) (2011) 1071–1080.

[10] H. C. Purchase, Metrics for graph drawing aesthetics, Journal of Visual Languages and Computing 13 (5) (2002) 501–516.

[11] C.-C. Lin, W. Huang, W.-Y. Liu, S. Tanizar, S.-Y. Jhong, Evaluating esthetics for user-sketched layouts of clustered graphs with known clustering information, Journal of Visual Languages & Computing 37 (2016) 1–11.

[12] W. Huang, P. Eades, How people read graphs, in: Proceedings of the Asia-Pacific Symposium on Information Visualisation, Vol. 45, 2005, pp. 51–58.

[13] J. Abello, F. V. Ham, N. Krishnan, Ask-graphview: A large scale graph visualization system, IEEE Transactions on Visualization and Computer Graphics 12 (5) (2006) 669–676.

[14] S.-H. Hong, H. Nagamochi, Convex drawings of hierarchical planar graphs and clustered planar graphs, Journal of Discrete Algorithms 8 (3) (2010) 282–295.

[15] Y. J. Ko, H. C. Yen, Drawing clustered graphs using stress majorization and force-directed placements, in: Proceedings of the 20th International Conference Information Visualisation (iV2016), 2016, pp. 69–74.

[16] J. Ho, S.-H. Hong, Drawing clustered graphs in three dimensions, in: Proceedings of the 13th International Symposium on Graph Drawing (GD2005), Vol. 3843 of Springer LNCS, 2006, pp. 492–502.

[17] T. Dwyer, K. Marriott, P. J. Stuckey, Fast node overlap removal, in: Proceedings of the 13th International Symposium on Graph Drawing (GD2005), Vol. 3843 of Springer LNCS, 2006, pp. 153–164.

[18] E. R. Gansner, Y. Hu, Efficient node overlap removal using a proximity stress model., in: Proceedings of the 15th International Symposium on Graph Drawing (GD2007), Vol. 5417 of Springer LNCS, 2008, pp. 206–217.

[19] M. Bostock, V. Ogievetsky, J. Heer, D3 data-driven documents, IEEE Transactions on Visualization and Computer Graphics 17 (12) (2011) 2301–2309.

[20] K. J. Pulo, Recursive space decompositions in force-directed graph drawing algorithms, in: Proceedings of Asia-Pacific Symposium on Information Visualisation (APVis '01), 2001, pp. 95–102.

[21] H.-Y. Wu, S. Takahashi, C.-C. Lin, H.-C. Yen, Voronoi-based label placement for metro maps, in: Proceedings of the 17th International Conference on Information Visualisation (IV2013), 2013, pp. 96–101.

[22] P. Brivio, M. Tarini, P. Cignoni, Browsing large image datasets through Voronoi diagrams, IEEE Transactions on Visualization and Computer Graphics 16 (6) (2010) 1261–1270.

[23] K. A. Lyons, H. Meijer, D. Rappaport, Algorithms for cluster busting in anchored graph drawing, Journal of Graph Algorithms and Applications 2 (1) (1998) 1–24.

[24] E. R. Gansner, S. C. North, Improved force-directed layouts, in: Proceedings of the 6th International Symposium on Graph Drawing (GD1998), Vol. 1547 of Springer LNCS, 1998, pp. 364–373.

[25] W. Didimo, F. Montecchiani, Fast layout computation of clustered networks: Algorithmic advances and experimental analysis, Information Sciences 260 (2014) 185–199.

[26] H.-Y. Wu, S. Takahashi, C.-C. Lin, H.-C. Yen, A zone-based approach for placing annotation labels on metro maps, in: Proceedings of the 11th International Conference on Smart Graphics, Vol. 6815 of Springer LNCS, 2011, pp. 91–102.

[27] H.-Y. Wu, S. Takahashi, C.-C. Lin, H.-C. Yen, Travel-route-centered metro map layout and annotation, Computer Graphics Forum 31 (3) (2012) 925–934.

[28] H.-Y. Wu, S. Takahashi, D. Hirono, M. Arikawa, C.-C. Lin, H.-C. Yen, Spatially efficient design of annotated metro maps, Computer Graphics Forum 32 (3) (2013) 261–270.

[29] G. Stadler, T. Steiner, J. Beiglböck, A practical map labeling algorithm utilizing morphological image processing and force-directed methods, Cartography and Geographic Information Science 33 (3) (2006) 207–215.

[30] K. E. Hoff, III, J. Keyser, M. Lin, D. Manocha, T. Culver, Fast computation of generalized Voronoi diagrams using graphics hardware, in: Proceedings of SIGGRAPH '99, 1999, pp. 277–286.

[31] G. Taubin, Curve and surface smoothing without shrinkage, in: Proceedings of the 5th International Conference on Computer Vision (ICCV'95), 1995, pp. 852–857.

[32] G. Taubin, A signal processing approach to fair surface design, in: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95, 1995, pp. 351–358.

[33] U. Brandes, A faster algorithm for betweenness centrality, Journal of Mathematical Sociology 25 (2) (2001) 163–177.