

# Clutter-Aware Label Layout

Yu Meng, Hui Zhang, Mengchen Liu, Shixia Liu

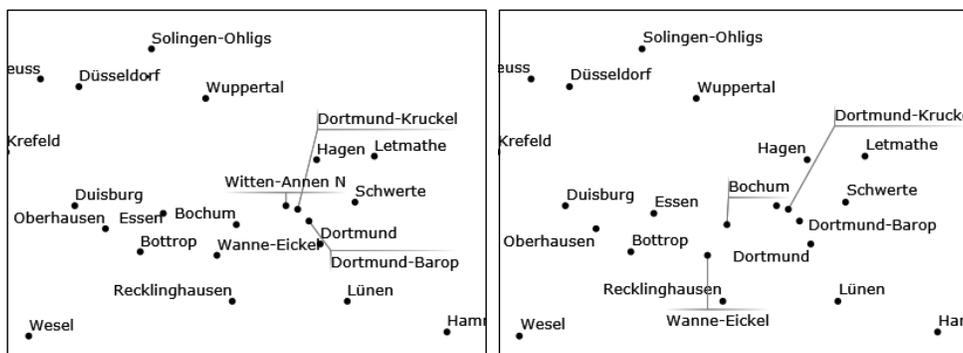


Fig. 1. A comparison of partial layouts of 366 “German railway stations.” The labeling results are generated on canvases with a resolution of  $2850 \times 3800$ . The left side shows particle-based labeling [18]. The right side is the result of our labeling pipeline with clutter control.

**Abstract**— A high-quality label layout is critical for effective information understanding and consumption. Existing labeling methods fail to help users quickly gain an overview of visualized data when the number of labels is large. Visual clutter is a major challenge preventing these methods from being applied to real-world applications. To address this, we propose a context-aware label layout that can measure and reduce visual clutter during the layout process. Our method formulates the clutter model using four factors: confusion, visual connection, distance, and intersection. Based on this clutter model, an effective clutter-aware labeling method has been developed that can generate clear and legible label layouts in different visualizations. We have applied our method to several types of visualizations and the results show promise, especially in support of an uncluttered and informative label layout.

**Index Terms**—Visual clutter, clutter measure, label placement.

## 1 INTRODUCTION

A high-quality label placement method without clutter is essential for effective information consumption and decision making. First, it can help illustrate and convey the visualized content by leveraging the descriptive nature of labels. Second, labels in illustrations allow users to better understand the association between graphical objects, thereby supporting their comprehension and analysis.

Over the past two decades, a great deal of effort has gone into automatic label layouts [18]. Most of these labeling approaches aim to create a label that is both readable and close to its element. Typically, a label is readable if it does not overlap with other graphical elements. Existing methods for offering an automatic label layout have achieved some success at addressing the non-overlapping issue. However, these label layout methods are inadequate in helping users to quickly understand the visualized data when the number of labels is large. This performance degradation is due to the difficulty in recognizing or searching for a label that is interfering with other surrounding labels or point-features, especially when the item spacing is small due to visual clutter caused by excess and/or disorganized visual elements. This issue can be solved simply by laying out only the important labels. However, this solution may lead to varying local density. Some regions in the generated layouts are sparse or even empty, while others are very dense due to the fact there are too many important point-features. To bridge the gap, the fundamental issue of decreased recognition

performance due to a cluttered label layout needs to be addressed.

There are two technical challenges that we believe are critical to generating an uncluttered label layout. The first challenge is how to define and estimate visual clutter in a visualization. Clutter is an important consideration in the design and development of information visualizations. Too many and/or disorganized graphical elements typically cause decreased recognition performance due to occlusion, the difficulty of recognizing elements and performing visual search [22].

Recently, researchers have proposed several metrics and models to measure visual clutter in an image. A question then naturally arises: can we view a visualization as an image and directly apply an image clutter measure method to the visualization? After careful study, we realized that the image clutter measure is not optimal for visualization measurement since visualizations have special layouts with interactive visual elements rather than static pixels. It is therefore important to understand what features, attributes, and factors are relevant to visual clutter in a visualization and how to measure it. The second challenge is how to generate an uncluttered label layout by considering a computational measure of clutter. When examining a visualization, the user will be overwhelmed if it provides too much content with clutter. On the other hand, if too little information is provided, the visualization is not cluttered. However, the user may have little to go on. As a result, it is also preferable to make a good tradeoff between informative visualization and uncluttered visualization.

To address these challenges, we propose a new label layout pipeline, in which we incorporate a computational measure of clutter into the existing label layout process. Instead of developing a new label layout algorithm, we propose a method that can improve a given label layout to produce clutter-free labeling for illustration. In particular, we adopt a state-of-the-art method, particle-based labelling [18], to place the labels in a visualization. Our method first preprocesses the input visualization to roughly estimate the clutter degree of each point-feature and then rank the related point-features according to the clutter degree and

- Yu Meng is with School of Software, Tsinghua University, E-mail: yumeng@cs.unc.edu.
- Hui Zhang is with School of Software, Tsinghua University, E-mail: hui.zhang@tsinghua.edu.cn.
- Mengchen Liu is with Tsinghua University, E-mail: liumc13@mails.tsinghua.edu.cn.
- Shixia Liu is with School of Software, Tsinghua University, E-mail: shixia@tsinghua.edu.cn.

IEEE Pacific Visualization Symposium 2015  
14–17 April, Hangzhou, China

U.S. Government work not protected by U.S. copyright

domain knowledge. In many visualizations, it is important to lay out as many labels as possible for the important point-features. This step will make sure that the important nodes have higher priorities. The proposed method formulates the clutter model according to four factors: confusion, visual connection, distance, and intersection. The uncluttered label layout is iteratively generated based on the point-feature ranking list and the clutter degree of the considered region.

By incorporating a computational measure of visual clutter into the label layout, our method offers two major technical contributions:

- **An effective clutter estimation model** that can measure the clutter degree of each label by leveraging four clutter factors: confusion, visual connection, distance, and intersection.
- **An improved clutter-aware label layout pipeline** that can generate an uncluttered label layout by considering the clutter degree in a visualization.

## 2 RELATED WORK

In the last two decades, a large number of clutter reduction techniques for information visualization have been developed [7]. For example, Bertini and Santucci [5] suggest a method of sampling and pixel displacement to reduce the clutter in 2D scatter plots. Frishman and Tal [8] present a physically-inspired method to reduce the clutter in graph layouts. However, the above methods do not explicitly define a formal model of visual clutter or exactly measure visual clutter in a visualization. Thus, the approaches are not clear enough as to what extent they reduce visual clutter and how they are able to do so.

To solve this problem, there are some initial efforts to define metrics to measure visual clutter. Each metric is usually specific to one particular visualization [14]. For instance, [4, 3] measure the clutter in scatter plots. [13, 14] measure the clutter in Euler diagrams. [20] measures the clutter in parallel coordinates, scatter plots, star glyphs, and dimensional stacking using different metrics. To the best of our knowledge, there is not yet a clutter model for label layouts. Since labels are generally used in a variety of visualizations, the model and corresponding metrics for measuring visual clutter in label layouts are very critical for understanding the content in a visualization.

On the other hand, in computer vision, a great deal of effort has gone into estimating the visual clutter of an image. Rosenholtz et al. present a feature congestion method to measure visual clutter [22]. They claim that feature congestion is one of the major causes of clutter, and therefore the level of feature congestion can be used to estimate the degree of clutter. In [23], Rosenholtz et al. introduce another clutter measure method called Subband Entropy measure. This method is based on the notion that the more “organized” an image is, the less cluttered it is. If an image is “well-organized,” the number of bits required for coding it will be small. Thus the bit number is a reasonable measure of visual clutter. More recently, Bravo et al. [6] use the proportionality constant of a power law to measure visual clutter. They claimed that when an image is segmented, the relationship between the number of regions and the scales of segmentation follows a power law. Generally, the more cluttered an image, the larger constant of proportionality the power law has. Berg et al. [25] propose a crowding model to measure visual clutter. They suggest that “*crowding appears to be the result of feature pooling, carried out by (weighted) integration fields with sizes proportional to retinal eccentricity*” [19]. Therefore, they simulate the process of feature pooling and use the amount of lost information as an indicator of visual clutter.

Although the above methods achieve some success in measuring the visual clutter of an image, they cannot be applied to a visualization effectively or efficiently. The major reason is that a visualization contains a lot of interactive visual elements rather than static pixels, and treating the display as an image fails to consider the relationships between visual elements.

Some other methods related to human cognition, perception, and understanding, have a close relationship with visual clutter. The visual clutter will lead to a degradation of performance at some task [22], which means the visual clutter is an obstacle to cognizing, perceiving, or understanding the visualizations. If we were able to understand how the human brain cognizes and processes the information displayed by the

Table 1. Frequently used notations and their corresponding explanations.

Notations	Explanations
$t_r$	Reasoning time, which is measured by the number of reasoning steps.
$b_m$	Memory burden, which is measured by the maximum number of visual elements that must be kept in the STM at the same time.
$C_{stm}$	The degree of confusion, which is equal to the weighted sum of $t_r$ and $b_m$ .
$C_{vc}$	The times that visual connection happens.
$d_v$	Visual distance.
$D_c$	Visual difficulty caused by visual connection.
$D_d$	Visual difficulty caused by visual distance.
$D_i$	Visual difficulty caused by intersection.
$D_{pp}$	The degree of visual difficulty in the perceptual process, equal to the sum of $D_c$ , $D_d$ , and $D_i$ .

visualizations and determine what may hinder human brains from doing this, we could detect and measure visual clutter and avoid it based on a clear theoretical foundation. Fortunately, researchers have made many achievements in understanding human cognition, perception, or comprehension in biology [2], psychology [15, 21, 19, 1], human computer interaction [17], and information visualization [9, 11, 26, 10, 16].

Motivated by the above research, we simulate the brain’s reasoning process when it tries to interpret a visualization full of labels and evaluate how difficult this process is. Then we combine this evaluation with traditional criteria to model an effective measure of visual clutter.

## 3 CLUTTER MODEL

Visual clutter makes it difficult to search for and understand information in a visualization. As a result, it is important to study what factors influence the clutter in a visualization. The classic models of perception and cognition contain three major components: the perceptual process (or sensory memory), short-term memory (STM) and long-term memory (LTM) [17, 10]. When a user views a visualization, the perceptual process encodes the visual information, such as position, shape, size, and color, and passes it to the STM. Then the STM decodes the information and understands it. If the user is trying to explore more implicit information behind the data, the process for forming the LTM will be triggered. Otherwise the cognitive process is finished. Forming the LTM is a complex mental process depending on many other factors such as one’s experience, basic knowledge, intelligence development, and so on, and it is beyond the reach of a visualization. A visualization mostly benefits the perceptual process and the STM. Therefore, the visual clutter of a visualization mostly affects the perceptual process and STM.

When users read labels, they face two basic tasks: searching for a given point-feature’s associated label, or searching for a given label’s associated point-feature. The complex tasks are usually organized by many basic tasks in different forms. To make the analysis easy and clear, in the rest of this paper when we talk about the process of human perception and cognition, we simply mean the process of performing basic tasks.

Next, we will analyze the causes of the clutter in the perceptual process and the STM, and introduce how to evaluate each of them. Then we present a computable clutter model that can effectively measure the degree of clutter in a visualization. Table 1 describes some notations that are useful for subsequent discussions.

### 3.1 Confusion

The first factor, **Confusion**, describes the degree of difficulty users experience when processing information in the STM. Before defining confusion, we use the basic task of “finding the associate point-feature of the label ‘Chevrolet’ (See Figure 2(a))” to illustrate how the STM works with a label layout.

We first assume that we have found the label “Chevrolet” in Figure 2(a) and it has been encoded in the STM. There are five point-features (Ⓓ, Ⓒ, Ⓔ, Ⓖ, Ⓢ) around “Chevrolet,” each of which might be its

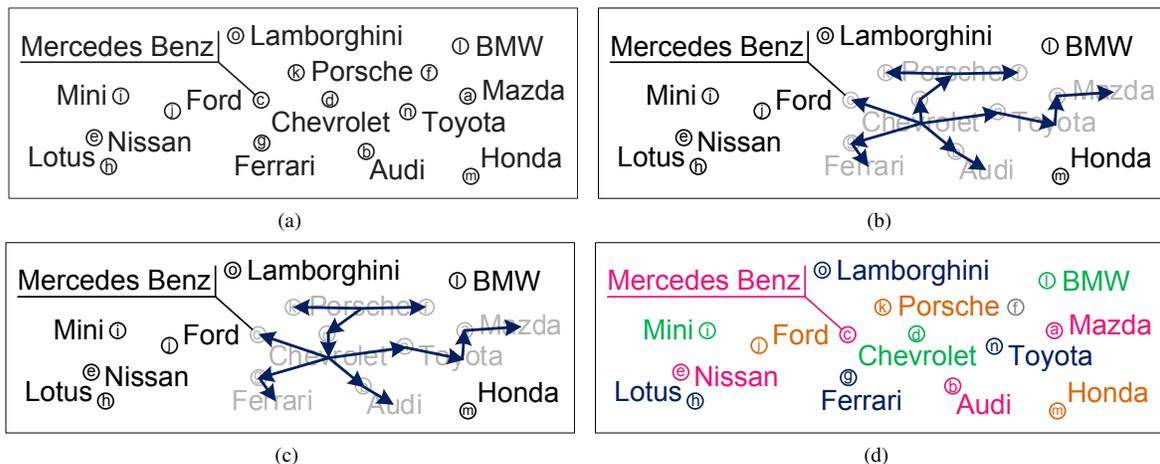


Fig. 2. Cognitive process. (a) An example label layout. Each point-feature is represented by a circle with a letter. (b) and (c) The reasoning process of recognizing a label. The arrows show the reasoning direction. (d) A pair of point-feature and its associated label with different colors.

associate point-feature. As a result, these five point-features are pushed into the STM and prepared for further checking one-by-one.

For point-feature ①, label “Porsche” is also next to it. To make it clear whether ① is associated with “Chevrolet” or “Porsche,” we need to put “Porsche” into the STM and check all the other point-features around it. However, both ④ and ① may be the point-feature associated with “Porsche,” and there is no other information we can use for further reasoning. Thus we fail to judge whether ① is the point-feature associated with “Chevrolet” based on the above analysis. Now we do not need to keep “Porsche,” ④, or ① in the STM any more, but we do need to remember that ① has been checked and has not been associated with any label.

Then we check point-feature ③. It is obvious that it has a guiding line, which connects its label. We can quickly conclude that ③ is not the point-feature associated with “Chevrolet,” even without following the guiding line to find “Mercedes Benz.”

Next, point-feature ② is checked. In addition to “Chevrolet,” “Ferrari” is also next to ②. So “Ferrari” is pushed into the STM. We then find that ② is the only point-feature around “Ferrari.” Thus we derive that ② is not the point-feature associated with “Chevrolet” but with “Ferrari.” Similarly, we can conclude that ⑤ is not the associated point-feature of “Chevrolet” but of “Audi.”

Finally, only ⑥ has not been checked. Around ⑥, there is label “Toyota” in addition to “Chevrolet,” so “Toyota” is pushed into the STM. “Toyota” can be the label associated with ⑥ or ④ and as a result, ④ is pushed into the STM. Then we find that ④ is the only point-feature around “Mazda,” so “Mazda” belongs to ④, and consequently “Toyota” belongs to ⑥. At this moment, “Mazda,” “Toyota,” and ④ no longer need to be remembered. What we need to remember is that ⑥ is not the point-feature of “Chevrolet.”

Now, we have checked all five point-features around “Chevrolet.” The result is that ① is the only point-feature that has not been associated with a label. Thus ① is the point-feature that “Chevrolet” belongs to.

The above example illustrates how humans reason during a basic task. The arrows in Figure 2(b) illustrate the trace of the reasoning. All the labels and point-features that are passed by the arrows were pushed into the STM. The longer the trace is, the more visual elements (point-features and labels) need to be kept in the STM, and the longer performance time is needed. What is worse, because the STM is limited, remembering too much new information will lead to forgetting previous information, and force people to repeat reasoning effort.

The label layout in Figure 2(a) is not good because the layout makes the association relationships very ambiguous. Users have to make an effort to comprehend and reason. This phenomenon is called **Confusion**. To measure it, we count the number of reasoning steps and the number of visual elements that are pushed into STM. The number of reasoning steps reflects the time cost  $t_r$  and the number of visual elements reflects the memory burden  $b_m$ .

As shown in Figure 2(b), the reasoning process can be described

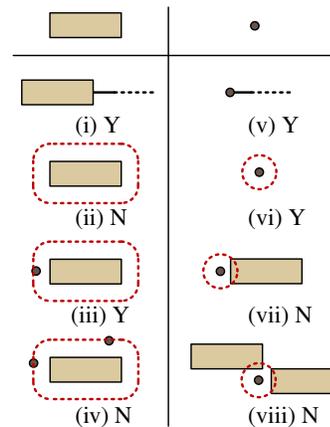


Fig. 3. Reasoning Cases. The left column shows cases of recognizing a label. The right column shows cases of recognizing a point-feature. “Y” means the visual element can be recognized directly, while “N” means it cannot be recognized. The area enclosed in a red line represents the neighborhood of a visual element.

as a directed graph, called the reasoning graph. The visual elements (point-features and labels) are nodes and the arrows are directed edges. The number of reasoning steps is equal to the number of directed edges. In this example,  $t_r = 13$ .

One necessary condition of reasoning a label is that there is no cycle in the directed graph. For example, in Figure 2(a), it is impossible to determine the point-features of “Lotus” and “Nissan” because there is a reasoning cycle in the cognitive process. To associate “Lotus” with ①, “Nissan” should be associated with ③ first, and vice versa. In this case, the number of reasoning steps is infinite.

In the cognitive process, the visual element in the STM changes from time to time. The memory burden  $b_m$  is the maximum number of visual elements that must be kept in STM at the same time. Since the reasoning graph has no cycle, it can be considered a tree, called a reasoning tree (See Figure 2(b)). The label to be reasoned is the root. The value of  $b_m$  is equal to the number of vertices incident to the root plus the height of the tree. In this example,  $b_m = 9$ .

To calculate the values of  $t_r$  and  $b_m$  of a label, the reasoning tree needs to be built first. If a cycle exists in the reasoning process, the tree building fails. Consequently  $t_r = +\infty$  and  $b_m$  is meaningless.

Figure 3 enumerates all the possible cases of associating the label and the point feature. In this figure, the left column associates the label with the point feature, while the right one associates the point feature with the label. In cases (i) and (v), the point-feature or the label has a guiding line, and thus they can be associated. Since some labels cannot be placed due to limited space, a label layout may have some point-features without labels (See point-feature ① in Figure 2(d)). Case (vi) reflects this situation. In case (iii), there is only one point-feature

within the label’s neighborhood, thus they are associated. While in case (vii), even though there is only one label within the point-feature’s neighborhood, it is irrational to associate them. This is because in a label layout, a point-feature is allowed to have no label and the label may belong to another point-feature. If there are more than one point-features (case (iv)) or labels (case (viii)) in the neighborhood of the visual element being checked, it cannot be associated.

However, the reasoning tree cannot guarantee that the label can always be associated. For example, in Figure 2(c), if we use “Porsche” as the root, a reasoning tree is built. However, because both  $\textcircled{K}$  and  $\textcircled{I}$  have not been recognized, “Porsche” may belong to either of them. A leaf in a reasoning tree has two states: can or cannot be recognized directly. We define a leaf that cannot be recognized directly as an invalid-leaf and a subtree containing invalid-leaves as an invalid-subtree. Now we can conclude that the sufficient and necessary condition that the root can be recognized by the reasoning process is that it contains at most one direct invalid-subtree.

To sum up, the degree of confusion is measured by

$$C_{stm} = \begin{cases} \omega_{rst}r + \omega_{mb}b_m, & \text{if the label can be recognized} \\ +\infty, & \text{otherwise.} \end{cases} \quad (1)$$

We empirically set  $\omega_{rs} = 0.3$  and  $\omega_{mb} = 1$  for all our results.

### 3.2 Visual Difficulty

We use **visual difficulty** to describe the difficulties caused by visual distracters in the perceptual process. The perceptual process is fast and unconscious and a visualization will benefit greatly if it takes advantage of the perceptual process. Figure 2(d) gives a good example of making use of the perceptual process to accelerate perception and cognition. Because color and letter can be comprehended in parallel, if we use color to distinguish different pairs of point-features and labels, the association relationship between them will be recognized without any further reasoning. However, using color is not always an effective way to accelerate recognition because color is usually used to indicate other information, such as priority, categorization, etc. Furthermore, if there are too many label/point-feature pairs, color scalability will become an issue. As a result, in this paper, without loss of generality, we assume color cannot be used to help associate point-features with labels.

Here we consider three distracters impeding the perceptual process. They are **visual connection**, **visual distance**, and **intersection**. Assuming that  $D_c$ ,  $D_d$ , and  $D_i$  indicate the visual difficulties caused by each distracter, respectively, the degree of visual difficulty in the whole perceptual process is

$$D_{pp} = D_c + D_d + D_i. \quad (2)$$

Next, we will explain why each distracter impedes the perceptual process, and introduce how to detect or measure visual difficulties.

#### 3.2.1 Visual Connection

Visual connection is used by Eduard Imhof [12] to describe the poor placement of names on maps. It happens when the names are too close or one name is split by another. In a label layout, visual connection is the phenomenon in which the labels are aligned closely. For example, there are two labels in Figure 4(a). However, they are ambiguous. They may be “graduate school” and “board,” or they may be “graduate” and “school board.” The reason for this ambiguity is that people cannot tell the difference between the gap between words and the gap between labels. The ambiguity can be removed as long as the labels are further apart (Figure 4(b)), or have an obviously vertical gap (Figure 4(c)).

To detect visual connection between two labels, we check their horizontal gap  $w_h$  and vertical gap  $\Delta_v$ . In our experiments, assuming the width of a space character is  $w_s$ , the height of the label is  $h_l$ , if  $w_h \leq 3w_s$ , and at the same time  $\Delta_v \leq 0.2h_l$ , the two labels are considered visually connected. The visual difficulty caused by visual connection can be estimated by

$$D_c = \omega_{vc}C_{vc}, \quad (3)$$

where  $C_{vc}$  is the number of times that visual connection happens. We empirically set  $\omega_{vc} = 2$ .

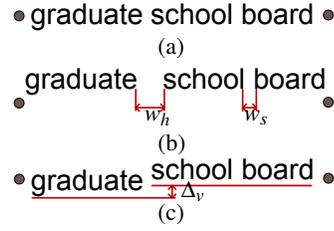


Fig. 4. Visual Connection. In (a) the two labels are visually connected. In (b), since the label’s horizontal gap  $w_h$  is much larger than  $w_s$ , the width of a space character, they are distinguished easily. In (c), although the labels are close to each other, the visual connection is broken due to the vertical gap between the two labels.

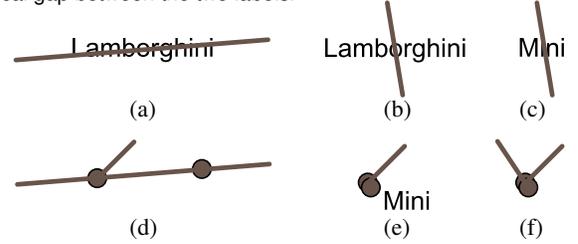


Fig. 5. Intersections between labels and guiding lines.

#### 3.2.2 Visual Distance

In a visualization, labels are expected to be close to their associated point-features. The longer the distance between a label and its point-feature, the more time it takes to associate them, even with the help of a guiding line.

What we care about is how long the eyes move in the association process. Thus we define visual distance  $d_v$ , which is the length by the eye movement.

Visual distance depends on whether the label has a guiding line. If there is no guiding line, the eye movement route will be a straight line, because humans tend to minimize energy cost unconsciously when finishing a task. In this situation, the visual distance is the minimum distance from a point-feature to its label boundary. On the other hand, if the label has a guiding line, the eye’s focus tends to follow the guiding line. Thus the visual distance is equal to the length of the guiding line.

The visual difficulty caused by the visual distance is estimated by

$$D_d = \omega_{vd}d_v. \quad (4)$$

$\omega_{vd}$  is empirically set as 0.05 in our experiments.

#### 3.2.3 Intersection

In our work, we do not consider label-label intersections or label-point intersections, because currently almost all layout methods are able to avoid them. What we focus on are the intersections caused by a guiding line. A guiding line may intersect with irrelevant point-features, labels, or other guiding lines. These intersections make the guiding lines difficult to follow. Intuitively, the number of intersections is an effective indicator of this difficulty, but it is not enough. We will consider two more complicated situations.

We first consider a situation where a guiding line intersects with labels. As shown in Figure 5, the label in (a) is less readable than the one in (b) because there are more covered letters in (a) than (b). The number of covered letters in (b) is equal to that in (c), but “Lamborghini” is much easier to recognize than “Mini” because there are more uncovered letters in (b) to help people guess the covered part.

A conclusion drawn from the above example is that the higher percentage of covered letters, the more difficult it is for people to read. Thus we use  $l_i^2/l_d^2$  to measure the difficulty caused by covering, where  $l_i$  is the length of the parts of the guiding line falling in the label region and  $l_d$  is the distance between the two furthest points on the label boundary. For example, if the label shape is a rectangle,  $l_d$  is the length of the diagonal.

Next, we study a situation where a guiding line intersects with point-features. In Figure 5(d), the guiding line passes two irrelevant

point-features, leading to ambiguity as to where it starts. This situation needs to be avoided. However, if the point-features are overlapping, as shown in 5(e) and 5(f), such an intersection cannot be avoided unless the guiding line and its connected label are removed. In this situation, it is the overlapping of point-features but not the intersection that is likely the cause of ambiguity. Since the positions of point-features are given, we ignore this situation when considering the intersection between guiding lines and point-features.

To sum up, for a label with a guiding line, the visual difficulty caused by an intersection is measured by

$$D_i = \omega_{ig}T_{ig} + \omega_{il}T_{il} + \omega_{ip}T_{ip} + \omega_{ic} \sum \frac{l_i^2}{l_d^2}, \quad (5)$$

where  $T_{ig}$ ,  $T_{il}$  and  $T_{ip}$  are the times the guiding line intersects with other guiding lines, irrelevant labels, and point-features. The first three terms in Equation 5 reflect the difficulty of following the guiding line and the last term reflects the difficulty of reading the label. We empirically set  $\omega_{ig} = 1$ ,  $\omega_{il} = 1$ ,  $\omega_{ip} = 10$ , and  $\omega_{ic} = 10$ .

### 3.3 Clutter Metric

Considering both the confusion in the STM and the visual difficulties in the perceptual process, the 2D clutter vector of a label is defined as

$$\vec{C} = (D_{pp}, C_{stm}). \quad (6)$$

This metric is a local one. It describes the difficulty of performing a basic task: finding the associated point feature of a given label. Comparing this to the global metric, the advantage of a local one is that it can guide a label layout method in more detail. We will introduce how this works in the next section.

## 4 IMPLEMENTATION

The purpose of measuring visual clutter is to generate better label layouts, which are much clearer and easier for people to understand. In this section, we will introduce how to combine our clutter model with the existing label layout method and how the clutter model controls the labeling process. Particularly, our approach contains two steps: preprocessing and labeling.

### 4.1 Preprocessing

The preprocessing step ranks the point-features according to their importance and the density of the regions where the labels will be placed. We assume that the importance is predefined in this step. Here we focus on how to estimate the density. Before all the point-features are labeled, it is extremely difficult to evaluate the real density precisely. The density is defined as the number of labels covering the unit area. Fortunately, we do not need to know the real density; what we need is a rough but reasonable estimate that can be used to determine a better order for the labeling. Thus, a probability model is employed to roughly predict the density.

To simplify the computation, we make the following assumptions. First, all the point-features are particles and do not occupy any space. Second, all the labels will be placed next to their associated point-features. Third, no conflict will be managed, which means labels can overlap with each other or with point-features.

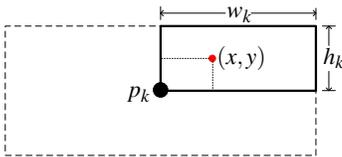


Fig. 6. An example of placing a label. The area bounded by the dash lines contains the possible position to place the label.

As shown in Figure 6, given a position  $(x, y)$  and a point-feature  $p_k(x_k, y_k)$ , the probability that this position will be covered by label  $p_k$

is

$$P_k(x, y) = \begin{cases} \frac{(w_k - |x_k - x|)(h_k - |y_k - y|)}{w_k h_k}, & \text{if } |x_k - x| \leq w_k \text{ and } |y_k - y| \leq h_k \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Then the density at this position can be estimated as

$$D(x, y) = \sum_{k=1}^n P_k(x, y), \quad (8)$$

where  $n$  is the number of point-features.

The density at the position of each point-feature can be estimated by Equation 8. However, such local density cannot be directly used to rank the point-features because it ignores the label conflicts. For example, the point-features in Figures 7(a) and (b) have the same local density according to Equation 8. However, the label layout in (a) has a higher density because the labels probably overlap. To overcome this problem, the conception of mean density is introduced. The mean density of a point-feature  $p_k$  is defined as:

$$\bar{D}(p_k) = \frac{1}{4w_k h_k} \int_{x-w_k}^{x+w_k} \int_{y-h_k}^{y+h_k} D(x, y) dx dy. \quad (9)$$

For the point-features with the same importance, they are ranked in the order of decreasing mean density values.

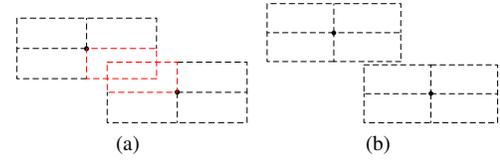


Fig. 7. Possible labels conflict. The labels in (a) may overlap, but the labels in (b) will not.

### 4.2 Labeling

In this step, the label placement and clutter measure operate alternately to derive a high-quality label layout without clutter. In our implementation, we employ a state-of-the-art label layout method, the particle-based method [18], to place labels. The particle-based labeling method labels point-features one-by-one. Given a feature-point, the particle-based method first tries to place the label at its closest neighbor. If the label overlaps with other visual elements, the labeling result is rejected and a farther position is checked. This process repeats until a non-overlapping position is found. If all the available positions result in overlaps, the feature-point is declared to be unlabeled. After a label is placed, clutter may be introduced by the label itself, the labels around it, or the labels intersecting with the newly added guiding line. The clutter metric proposed in Section 3.3 is employed to measure the clutter degree. If the result shows that the temporary labeling is more cluttered than the expectation, the label placement module will check another available position. Otherwise the temporary layout is accepted.

In the clutter measure module, we use a set of thresholds to measure whether the clutter degree is acceptable. However, the acceptability may differ for different users. Even for the same user, it may change for different tasks. For example, an expert may tend to use a label layout with more detailed information in spite of some clutter, but a novice may prefer to use a much clearer label layout with less information. Thus we provide an interactive controller to allow users to tune these thresholds, and consequently control the degree of clutter. Specifically, we use a variable  $V_c \in [0, 1]$  to control clutter. As  $V_c$  increases, the labels become more cluttered. The threshold of  $C_{stm}$  is  $60V_c$ ; the threshold of  $D_i$  is  $4V_c$ ; the threshold of  $D_d$  is  $10V_c h_l$ , where  $h_l$  is the height of the label; and the threshold of  $D_c$  is 0, which means visual connection must be avoided.

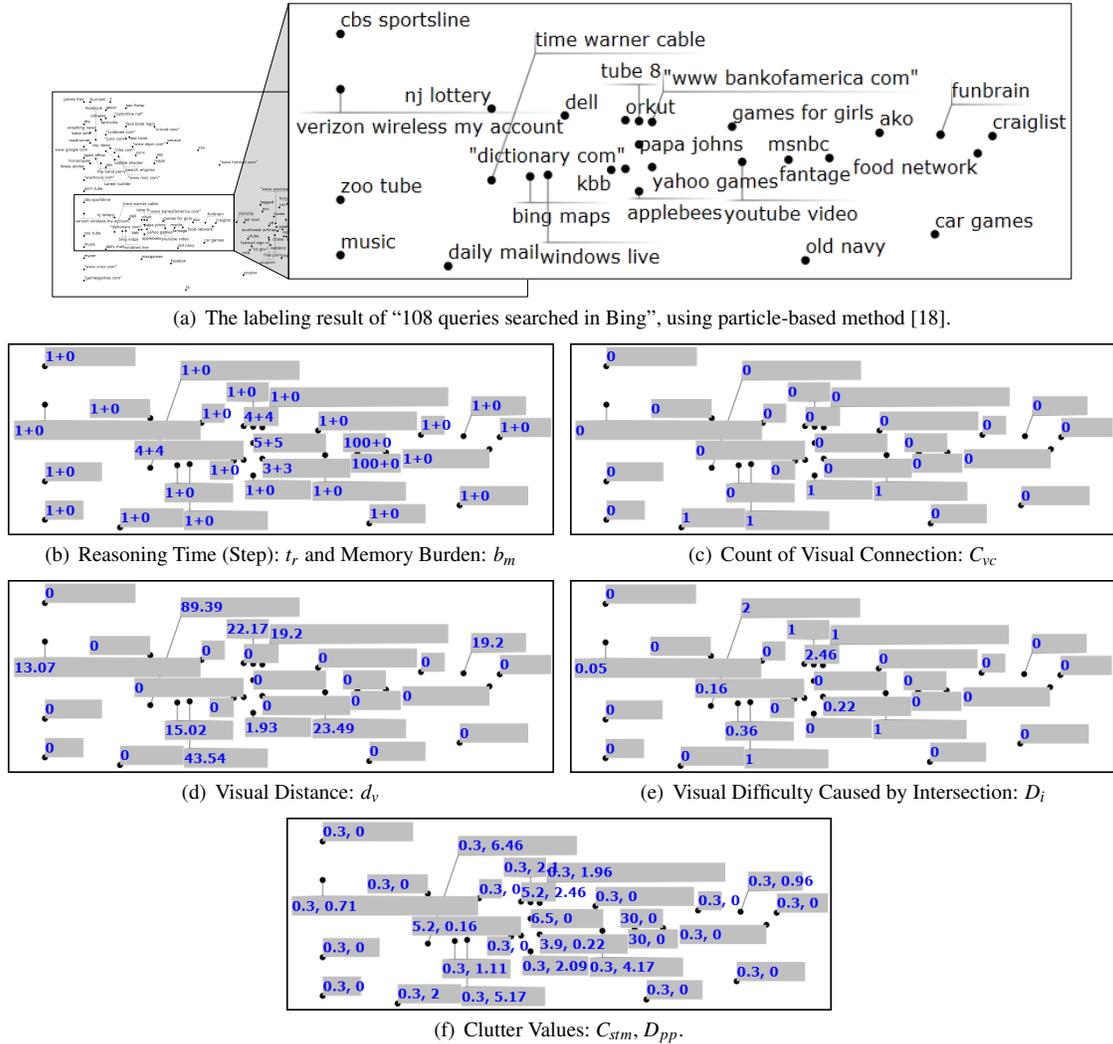


Fig. 8. Clutter measure with different metrics.

## 5 EXPERIMENTS

We implemented the clutter-aware labeling using C# and Silverlight. All the experiments were performed on a PC with a 2.00GHz Pentium(R) Dual-Core CPU and 3GB of RAM.

Figure 8 shows an example measured by our clutter metrics, including reasoning time  $t_r$ , memory burden  $b_m$ , the number of visual connection  $C_{vc}$ , visual distance  $d_v$ , the visual difficulty caused by intersection  $D_i$ , and the 2D visual clutter values. The data used in this experiment are a collection of queries searched for in Bing. The point-features are placed based on their importance (y-axis) and the time of the search (x-axis). The labels were placed by the particle-based method [18]. This example shows the effectiveness of our clutter metrics. For example, in Figure 8(a), the labels “msnbc” and “fantage” cannot be associated with their point-features due to the reasoning circle, so their reasoning steps shown in Figure 8(b) are both “100,” which means infinity in the implementation of our clutter model. Another example is that “applebees” is visually connected with “youtube video” (Figure 8(a)), thus the number of visual connections are both “1” (8(c)). Figure 8(f) shows the 2D visual clutter value of each label. The higher the clutter value of a label, the more difficult it is to associate it with its point-feature.

To verify and evaluate the effectiveness of our clutter control mechanism, we compared the results produced by the particle-based labeling method [18] with and without clutter control. The particle-based labeling method can place labels both adjacently and distantly, thus choosing this labeling method as a base can reflect the effectiveness of our clutter control mechanism in these two aspects. Here we used the

particle-based labeling method with the parameters mentioned in [18].

The comparisons were performed on both a random data set and several benchmark data sets. The random data was generated as follows: the positions of point-features were generated randomly within the display area and the labels were sequences of random Latin letters in lower case. Each sequence can be a random length from 1 to 16. The Latin letters were in “Lucida Sans Unicode” font of size “13.” The benchmark data sets employed here were “Tourist Shops in Berlin,” “1041 American Cities,” and “German Railway Stations.”

As shown in Table 2, we compared the results in the following aspects: the number of placed labels, the number of labels that can be recognized, the number of labels with confusion degrees larger than 6.5, the number of labels with visual difficulty values larger than 10, the mean value of the confusion degree, and the mean value of visual difficulty. Based on our experience, if the confusion degree of a label is larger than 6.5, it is hard to associate it with its point-feature; if the visual difficulty value of a label is larger than 10, the readability of the layout is much lower. Thus, in these comparisons, the number of labels that can be recognized is actually the number of labels whose confusion degrees are no more than 6.5. The statistics in Table 2 show that with a fixed resolution ( $2652 \times 1440$ ) the quality of labeling results without clutter control becomes obviously worse as the number of point-features increase. Specifically, more seriously cluttered ( $C_{stm} > 6.5$  or  $D_{pp} > 10$ ) labels appear and both  $\bar{C}_{stm}$  and  $\bar{D}_{pp}$  increase. On the other hand, the labeling results with clutter control almost prevent the production of seriously cluttered labels and  $\bar{C}_{stm}$  is kept around 0.3. Although with the clutter control, the labeling method may place fewer point-features, the number of labels that can be recognized is larger.

Table 2. Verification and evaluation of the effectiveness of the clutter control mechanism.  $N_{pf}$  represents the number of point-features.  $N_l$  is the number of point-features that were labeled.  $N_r$  is the number of labels that can be recognized.  $\bar{C}_{stm}$  is the mean value of the confusion degree and  $\bar{D}_{pp}$  is the mean value of visual difficulty. “W” and “WO” encode the particle-based labeling with and without clutter control, respectively.

Data Set	Resolution	$N_{pf}$	$N_l$		$N_r$		$N_l(C_{stm} > 6.5)$		$N_l(D_{pp} > 10)$		$\bar{C}_{stm}$		$\bar{D}_{pp}$	
			WO	W	WO	W	WO	W	WO	W	WO	W	WO	W
Random	$2652 \times 1440$	200	200	200	200	200	0	0	0	0	0.45	0.30	0.02	0.00
Random	$2652 \times 1440$	400	400	400	387	400	13	0	0	0	1.49	0.30	0.16	0.05
Random	$2652 \times 1440$	600	600	600	572	600	28	0	1	0	1.60	0.30	0.20	0.17
Random	$2652 \times 1440$	800	800	793	756	793	44	0	4	0	2.06	0.30	0.33	0.28
Random	$2652 \times 1440$	1000	1000	972	936	972	64	0	12	0	2.10	0.30	0.58	0.53
Shops	$12160 \times 6240$	357	357	346	333	346	24	0	4	0	2.45	0.30	0.63	0.43
Stations	$7216 \times 3744$	366	366	366	347	366	19	0	0	0	1.89	0.30	0.14	0.18
Cities	$7216 \times 3744$	1041	1041	1041	1001	1041	40	0	0	0	1.46	0.30	0.09	0.07

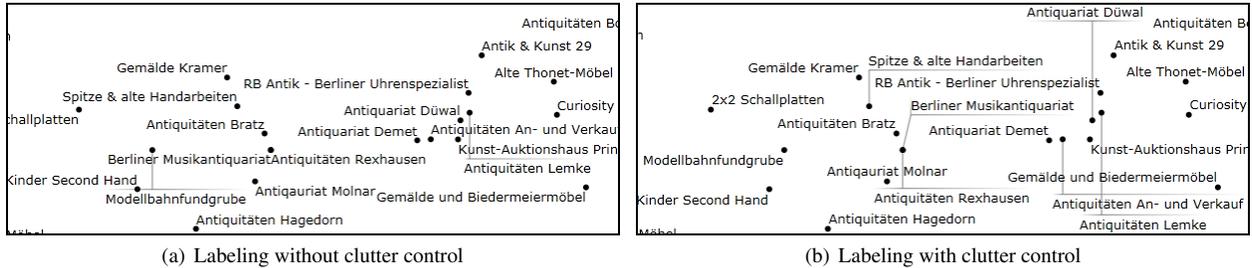


Fig. 9. Tourist Shops in Berlin. This data set contains 357 point-features. Here we only show a portion of the layout results.

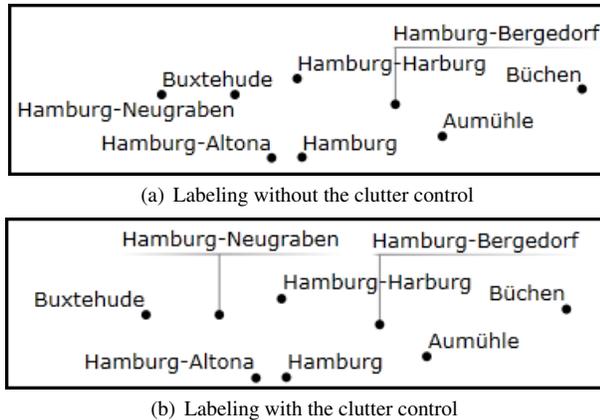


Fig. 10. German Railway Stations. Part of the layout results is shown.

Figure 9 compares the partial layout results on the benchmark data set “Tourist Shops in Berlin.” Without clutter control, the labels “Berliner Musikantiquariat” and “Antiquitäten Rexhausen” have terrible visual connection, while with clutter control, this situation is avoided. Additionally, with clutter control, it is much easier to associate the label “Antiquitäten An-und Verkauf” with its point-feature.

Figure 1 partially shows the comparison with data set “German Railway Stations.” The result with the clutter control mechanism is more readable. Figure 10 shows a part of the comparison. Without clutter control, it is hard to distinguish which point features the labels “Buxtehude” and “Hamburg-Neugraben” label because there is a reasoning circle. The clutter control mechanism can remove such a reasoning circle.

We also measured the performance of labeling methods with and without clutter control. The measurement results are given in Table 3. It can be seen that the time of the labeling with the clutter control is always about 15 times as long as that of particle-based labeling method. Two main factors contribute to the increased time consumption. First, in order to measure the clutter degree, two time-consuming operations, range-queries and collision-detections, are performed. We employ  $R^*Tree$  to accelerate these processes, but maintaining such a

data structure still needs a large amount of computation.  $R^*Trees$  are tree data structures used for spatial access methods, i.e., for indexing multi-dimensional information. The key idea of the  $R^*Tree$  is to group nearby objects and represent them with a minimum bounding rectangle in the next level of the tree. The bounding rectangles are used to decide whether or not to search inside a subtree. The  $R^*Tree$  supports quickly inserting and removing elements, and the search complexity is at worst  $O(M \log_M N)$ , where  $M$  ( $=4$ , in our experiments) is the maximum number of elements in each node. Second, with the clutter control mechanism, many more point-features are labeled distantly in order to avoid confusion, and the distant labels decrease the performance of particle-based labeling to some extent.

We conducted a user study to evaluate the effectiveness of our clutter model. Specifically, we applied a particle-based algorithm with and without clutter control to generate 10 label layouts. Five layouts were generated using the algorithm with clutter control and five were generated using the algorithm without clutter control. Two tasks were designed to evaluate the effectiveness of the developed visual clutter model: 1) associate a label with a given point-feature; 2) associate a point-feature with a given label.

We set up a website to run the user study. The participants’ answers and the completion time were recorded. In the user study, we recruited 33 participants and rejected participants whose answer accuracy was less than 50%. Answers from 30 participants were finally accepted.

Table 4 shows the final results, including the mean value and standard deviation of the answer accuracy and the completion time. The participants did better with the layout results generated by the algorithm with clutter control. The results also indicate that the participants’ performance was consistent with our clutter model. We performed a t-test on the users performance data and the results further demonstrated that the method with clutter control outperformed the method without clutter control significantly, with p-value 0.0043 ( $<<0.05$ ) for accuracy and p-value 0.03 ( $<0.05$ ) for the completion time.

## 6 CONCLUSION

We have proposed an effective clutter model for label layouts. The metrics consist of the confusion in the STM and the visual difficulty in the perceptual process. For confusion in the STM, the number of reasoning steps (time) and the memory burden are measured based

Table 3. Performance Measurements.  $N_l$  is the number of point-features that were labeled.  $N_r$  is the number of labels that can be recognized.  $\bar{C}_{stm}$  is the mean value of the confusion degree and  $\bar{D}_{pp}$  is the mean value of visual difficulty. “W” and “WO” encode the particle-based labeling with and without clutter control, respectively.

$N_{pf}$	Resolution	$N_l$		$N_r$		$\bar{C}_{stm}$		$\bar{D}_{pp}$		time(s)	
		WO	W	WO	W	WO	W	WO	W	WO	W
1000	3000 × 3000	1000	1000	976	1000	1.08	0.30	0.09	0.07	0.02	0.11
3000	6000 × 6000	3000	3000	2943	3000	1.00	0.30	0.05	0.03	0.02	0.37
6000	6000 × 6000	6000	5998	5729	5998	1.70	0.30	0.20	0.15	0.10	1.44
6000	9000 × 9000	6000	6000	5877	6000	0.96	0.30	0.04	0.02	0.06	0.94
7000	9000 × 9000	7000	7000	6870	6000	0.97	0.30	0.06	0.03	0.08	1.26
8000	9000 × 9000	8000	8000	7803	8000	1.14	0.30	0.08	0.05	0.16	1.57
9000	9000 × 9000	9000	9000	8766	9000	1.67	0.30	0.10	0.06	0.23	2.19

Table 4. The accuracy of answers and the completion time using the labeling methods with and without clutter control.

Method	Without clutter control	With clutter control
Accuracy (%)	60.7±29.6	94.7±8.4
Time (s)	17.7±27.6	11.0±14.4

on a reasoning tree that simulates how the human brain operates. For the visual difficulty in the perceptual process, the visual connection, the visual distance, and the intersection are considered. We have also proposed an approach to enhance existing labeling methods with our clutter control mechanism. Experiments on real-world data sets show that our clutter-aware labeling pipeline is able to produce clear and legible label layouts.

Our current clutter-aware labeling pipeline needs to be improved in some respects in the future. First, the performance should be improved. We aim at designing a faster way to detect collisions and perform range-query, which can greatly accelerate the process of clutter estimation. Second, some special visual elements or criteria should be considered for adding into the clutter model for special visualizations. For example, when we place labels on subway maps, the label position consistency [24] should be estimated.

## ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 61373070) and Tsinghua University Initiative Scientific Research Program (Grant No. 2012Z02170).

## REFERENCES

- [1] A. L. Alter, D. M. Oppenheimer, N. Epley, and R. N. Eyre. Overcoming intuition: metacognitive difficulty activates analytic reasoning. *Journal of experimental psychology General*, 136(4):569–576, 2007.
- [2] S. Baldassi, N. Megna, and D. C. Burr. Visual clutter causes high-magnitude errors. *PLoS Biol*, 4(3):e56, 02 2006.
- [3] E. Bertini. Is it darker? improving density representation in 2d scatter plots through a user study. In *Proc. of SPIE Conference On Visualization and Data Analysis*, January 2005.
- [4] E. Bertini and G. Santucci. Quality metrics for 2d scatterplot graphics: Automatically reducing visual clutter. In A. Butz, A. Krger, and P. Olivier, editors, *Smart Graphics*, volume 3031 of *Lecture Notes in Computer Science*, pages 77–89. Springer, 2004.
- [5] E. Bertini and G. Santucci. Improving 2d scatterplots effectiveness through sampling, displacement, and user perception. *Information Visualisation, International Conference on*, 0:826–834, 2005.
- [6] M. Bravo and H. Farid. A scale invariant measure of image clutter. *Journal of Vision*, 1(8):1–9, 2008.
- [7] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13:1216–1223, November 2007.
- [8] Y. Frishman and A. Tal. Uncluttering graph layouts using anisotropic diffusion and mass transport. *IEEE Transactions on Visualization and Computer Graphics*, 15:777–788, September 2009.
- [9] C. Healey and J. Enns. Attention and visual memory in visualization and computer graphics. *IEEE Trans Vis Comput Graph*, 2011.
- [10] W. Huang, P. Eades, and S.-H. Hong. Measuring effectiveness of graph visualizations: a cognitive load perspective. *Information Visualization*, 8:139–152, June 2009.
- [11] J. Hullman, E. Adar, and P. Shah. Benefitting infovis with visual difficulties. *IEEE Transactions on Visualization and Computer Graphics*, 17:2213–2222, Dec. 2011.
- [12] E. Imhof. Positioning names on maps. *Cartography and Geographic Information Science*, 2(2):128–144, Oct. 1975.
- [13] C. John. Measuring and reducing clutter in euler diagrams. *Electron. Notes Theor. Comput. Sci.*, 134:103–126, June 2005.
- [14] C. John, A. Fish, J. Howse, and J. Taylor. Exploring the notion of clutter in euler diagrams. In D. Barker-Plummer, R. Cox, and N. Swoboda, editors, *Diagrammatic Representation and Inference, 4th International Conference, Diagrams 2006, Stanford, CA, USA, June 28-30, 2006, Proceedings*, volume 4045 of *Lecture Notes in Computer Science*, pages 267–282. Springer, 2006.
- [15] S. M. Kosslyn. Understanding charts and graphs. *Applied Cognitive Psychology*, 3:185–225, July/September 1989.
- [16] Z. Liu and J. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16:999–1008, Nov. 2010.
- [17] J. Lohse. A cognitive model for the perception and understanding of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI ’91, pages 137–144, New York, NY, USA, 1991. ACM.
- [18] M. Luboschik, H. Schumann, and H. Cords. Particle-based labeling: Fast point-feature labeling without obscuring other visual features. *IEEE Transactions on Visualization and Computer Graphics*, 14:1237–1244, November 2008.
- [19] D. Pelli, M. Palomares, and N. Majaj. Crowding is unlike ordinary masking: distinguishing feature integration from detection. *J Vis*, 4(12):1136–69, 2004.
- [20] W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 89–96, Washington, DC, USA, 2004. IEEE Computer Society.
- [21] S. Pinker. *Artificial intelligence and the future of testing*, volume Artificial, chapter 4: A theory of graph comprehension, pages 73–126. Lawrence Erlbaum Associates, 1990.
- [22] R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. Feature congestion: A measure of display clutter. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 761–770, 2005.
- [23] R. Rosenholtz, Y. Li, and L. Nakano. Measuring visual clutter. *Journal of Vision*, 7(2):1–22, 2007.
- [24] J. Stott, P. Rodgers, J. C. Martinez-Ovando, and S. G. Walker. Automatic metro map layout using multicriteria optimization. *IEEE Transactions on Visualization and Computer Graphics*, 17:101–114, January 2011.
- [25] R. van den Berg, F. W. Cornelissen, and J. B. T. M. Roerdink. A crowding model of visual clutter. *Journal of Vision*, 9(4):1–11, 2009.
- [26] J. S. Yi, Y.-a. Kang, J. T. Stasko, and J. A. Jacko. Understanding and characterizing insights: how do people gain insights using information visualization? In *Proceedings of the 2008 conference on BEyond time and errors: novel evaluation methods for Information Visualization*, BELIV ’08, pages 4:1–4:6, New York, NY, USA, 2008. ACM.