Adaptive Zooming in Point Set Labeling

Sheung-Hung Poon[1,⋆] and Chan-Su Shin[2,⋆⋆]

[1] Department of Mathematics and Computer Science, TU Eindhoven,
5600 MB, Eindhoven, the Netherlands.
spoon@win.tue.nl

[2] School of Electronics and Information, HUFS,
Mohyun-myun, Yongin-si, Gyunggi-do, Korea.
cssin@hufs.ac.kr

Abstract. A set of points shown on the map usually represents special sites like cities or towns in a country. If the map in the interactive geographical information system (GIS) is browsed by users on the computer screen or on the web, the points and their labels can be viewed in a query window at different resolutions by zooming in or out according to the users' requirements. How can we make use of the information obtained from different resolutions to avoid doing the whole labeling from scratch every time the zooming factor changes? We investigate this important issue in the interactive GIS system. In this paper, we build low-height hierarchies for one and two dimensions so that optimal and approximating solutions for adaptive zooming queries can be answered efficiently. To the best of our knowledge, no previous results have been known on this issue with theoretical guarantees.

Keywords. Computational geometry, GIS, map-labeling, zooming.

1 Introduction

Point set labeling is a classical and important issue in the geographic information systems (GIS). An extensive bibliography about the map labeling can be found in [13]. The ACM Computational Geometry Impact Task Force report [5] identifies the label placement as an important research area. Nowadays, user interactivity is extremely crucial in such systems, especially for those systems available on the web. For the success of the interactivity and real-time navigation on maps in the system, the internal paradigm of the database needs to be carefully designed so that the system adjusts accordingly to satisfy the user requirements and efficiently answer the user queries.

Several aspects of the interactivity and adaption for GIS have been studied in [2, 3, 11, 16]. In [11, 9, 14], it is pointed out that the zooming operation in the

⋆ This research was supported in part by the Netherlands' Organisation for Scientific Research (NWO) under project no. 612-065-307.

⋆⋆ This research was supported in part by the Hankuk University of Foreign Studies Research Fund of 2005.

interactive GIS is an important issue. Petzold et al. [11] considered the problem of zooming the map by using a data structure called the reactive conflict graph. Its purpose is to minimize the dynamic query time after extensive preprocessing. At the preprocessing stage, they created a complete graph between any pair of points. Each edge of the graph stores the scaling ratio when the labels of the two points start to overlap. Firstly this process is definitely slow. Secondly at any specific zooming factor, this process cannot guarantee the size of the query output when comparing to the optimal size at that resolution. The obvious reason is that this data structure does not store any clue about the optimal solution at a specific resolution.

At any resolution, we consider the following problem. Given n distinct points $P = \{p_i : 1 \leq i \leq n\}$ in the plane, each p_i is associated with a constant number, say κ, of axis-parallel rectangular labels of unit height and of width ω_i such that p_i lies on the left boundary of its κ labels. The goal is to maximize the number of non-overlapping labels for P. We call this problem the *κ-fixed-position problem*. We note that the one dimensional version of this problem considers all the points of P lying on the x-axis. Even the 1-fixed problem in two dimensions is NP-complete [7] although Roy et al. [12] showed that a special variant of the one-fixed-position problem can be solved in $O(n \log n)$ time. Moreover, several variants of the above stated problem are proven to be NP-complete [7, 8, 10]. Agarwal, van Kreveld and Suri [1] showed that a 2-approximation of the κ-fixed-position problem can be computed in $O(n \log n)$ time, and a $(1+\epsilon)$-approximation of the problem for any $\epsilon > 0$ can be computed in $O(n \log n + n^{\frac{2}{\epsilon}-1})$ time. Chan [4] improved the running time for finding a $(1+\epsilon)$-approximation to $O(n \log n + n\Delta^{\frac{1}{\epsilon}-1})$, where $\Delta \leq n$ denotes the maximum number of labels a point lies inside. Moreover, several sliding versions of this problem were extensively studied in [15]. In this paper, we define the zooming problem properly and precisely, and the we build a low-height hierarchy for efficient adaptive queries with theoretically guaranteed output.

A *zooming* on a set of point means that while the point-to-point distances are scaled by a constant factor, the label sizes of the points remain fixed. The *zooming query within a rectangular query window W* is that given any zooming scale, we want to find the optimal solution for the κ-fixed-position solution for the labels completely inside the query window W. Instead of directly considering the zooming problem, we consider another equivalent problem. Now suppose we do not perform any zooming, meaning that we fix the point-to-point distances, and we instead scale the font-size of the label texts by a constant scaling factor. The *font-scaling query within a rectangular query window W* is that by applying any scaling factor on the font of the label texts, we want to find the optimal solution for the κ-fixed-position solution for those labels completely inside the query window W. It is clear that our original zooming problem is equivalent to the font-scaling problem. For the simplicity of notations, we will only consider the font-scaling problem for the rest of the paper. See Figure 1 for an example of labels in two different font sizes, in which optimal sets of labels are drawn with solid lines. We denote the scaling factor at a resolution γ by ρ_γ. The point

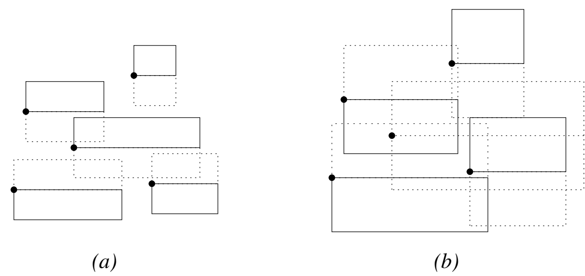

Fig. 1. An example when $\kappa = 2$. The font size/scaling factor in the right figure (b) is twice as larger as that in the left figure (a). Note that the collections of all the solid labels in both figures are optimal solutions.

set is said to be at the *coarser* resolution and has a *larger* scaling factor if it has a relatively larger font size; otherwise the point set is said to be at *finer* resolution and has a *smaller* scaling factor. We will use these two terminologies interchangeably to mean the same thing. For example, in Figure 1, the point set in Figure 1 (a) lies at a resolution α finer than the resolution β at which the point set in Figure 1 (b) lies. This means that the left figure has a scaling factor ρ_α smaller than the scaling factor ρ_β of the right figure.

In this paper, in order to achieve efficient adaptive zooming querying, the main backbone structure we build is a hierarchy of $O(\log n)$ levels, where each level represents one resolution, the lowest level has the finest resolution, and the resolutions become coarser and coarser when the levels in the hierarchy increase. On each level of the hierarchy, we store some data structures so that we can efficiently find the optimal or approximating solutions for adaptive zooming queries for the resolutions between any pair of consecutive levels. In one dimension, we build an $O(\log n)$-height hierarchy in $O(n \log n)$ time and in $O(n)$ space as stated in Theorem 1, and we can answer a zooming query for optimal solution efficiently as stated in Theorem 2. In two dimensions, we build an $O(\log n)$-height hierarchy in $O(n^2)$ time and in $O(n \log n)$ space as stated in Theorem 3, and we can answer a zooming query for approximating solution efficiently as stated in Theorem 4. In Section 2, we investigate the one-dimensional zooming problem. The two-dimensional version is studied in Section 3. Finally we conclude in Section 4.

2 Adaptive zooming in one dimension

Consider all the points p_i in P lie on x-axis. Each point p_i can choose to take any label say σ_i from its κ fixed-position choices. Label σ_i at point p_i can be represented as an interval $[p_i, p_i + \omega_i]$. Now finding the maximum number of non-overlapping labels is equivalent to finding the maximum independent set of the intervals $[p_i, p_i + \omega_i]$ where ω_i is the width of σ_i. The optimal solution can be computed using a greedy algorithm as described below. First, sort the right endpoints $p_i + \omega_i$ of the labels. Then select the label successively whose right endpoint is leftmost and moreover which does not intersect the label just selected. This algorithm runs in $O(n \log n)$ time. For any subset S of P, let $OPT_\gamma(S)$ denote the optimal solution at resolution γ obtained by running the above greedy algorithm on the labels at points in S. We denote $OPT_\gamma(P)$ by simply OPT_γ. Using the same greedy fashion, we describe a way to find the optimal solution at some resolution β by making use of the optimal solution at a finer resolution α in the following section. It will later serve as a main subroutine to build the hierarchy and to answer the zooming queries.

2.1 Computing $OPT_\beta(S)$ from OPT_α

Assume that the label at p_i is represented as intervals $[p_i, p'_{i,\alpha}]$ and $[p_i, p'_{i,\beta}]$ on x-axis at resolutions α and β respectively. It is clear that $p'_{i,\beta} > p'_{i,\alpha}$ as $\rho_\beta > \rho_\alpha$. Let $[q_k, q'_k]$ denote the kth label of OPT_α in the order from left to right. The algorithm to construct $OPT_\beta(S)$ in a greedy fashion by using OPT_α is presented in Algorithm *ComputeOPT*. In the algorithm, B_k denotes the subset of labels for points in S at resolution β such that the right endpoints (in sorted order) of the labels lie inside the interval $[q'_k, q'_{k+1})$ for some $1 \leq k \leq |OPT_\alpha|$. Note that B_k includes $[q_k, q'_k]$ itself. We assume $q'_{|OPT_\alpha|+1} = \infty$. Moreover, σ denotes the most recently selected label in the solution $OPT_\beta(S)$ by the algorithm.

Algorithm *ComputeOPT*(S)
Input. OPT_α and a set $S \subset P$.
Output. $OPT_\beta(S)$.
1. **for** each label $[p_i, p'_{i,\beta}]$ for points in S,
2. **do** Put $[p_i, p'_{i,\beta}]$ into B_k if $p'_{i,\beta} \in [q'_k, q'_{k+1})$ for some k.
3. Initialize $OPT_\beta(S) = \emptyset$, and $\sigma = \mathbf{nil}$.
4. **for** each B_k by incrementing k iteratively,
5. **do** Select the label σ' from B_k such that σ' does not overlap σ and has the leftmost right endpoint among the labels of B_k.
6. Put σ' into $OPT_\beta(S)$.
7. Set $\sigma = \sigma'$.

The idea is simply that as all the labels in B_k intersect q'_k, there is at most one label in B_k can be selected to put into $OPT_\beta(S)$. Note that binary search is used to put $[p_i, p'_{i,\beta}]$ into B_k in the first for-loop. The algorithm runs in $O(|S| \log |OPT_\alpha|)$ time.

Lemma 1. *ComputeOPT*(S) *computes the optimal solution* $OPT_\beta(S)$ *at resolution* β *by making use of* OPT_α *in* $O(|S| \log |OPT_\alpha|)$ *time.*

2.2 Building $O(\log n)$-height hierarchy

At each level of the hierarchy, we store the optimal solution at the resolution of the current level. The lowest level corresponds to the finest resolution, at which no labels can overlap, and the highest level corresponds to the coarsest resolution, at which the optimal solution has only a constant size. Between any pair of consecutive levels with resolutions α, β where $\rho_\beta > \rho_\alpha$, we have to determine a scaling factor $\rho = \rho_\beta/\rho_\alpha$ so that the size of the optimal solution $|OPT_\beta|$ drops significantly, say by a constant factor, from $|OPT_\alpha|$. If this can be done for each pair of consecutive levels, it results in an $O(\log n)$-height hierarchy. We show below how this can be done.

Building the lowest level. We need to decide a resolution, at which no labels overlap. Observe that $|p_{i+1} - p_i|/\omega_i$ is the minimum scaling ratio for the label at p_i to intersect the label at p_{i+1}. Let $\rho = \min_i\{|p_{i+1} - p_i|/\omega_i\}$. If we scale all labels by a factor a little smaller than ρ, no labels can overlap anymore. Thus we set $\rho - \epsilon$ (where $\epsilon > 0$ is small) to be the scaling factor for the lowest level. This step takes $O(n \log n)$ time as we need to first sort the points in P.

Building one level higher. As we have just discussed, to construct a level higher with resolution β from a level with finer resolution α, we need to decide a scaling factor $\rho = \rho_\beta/\rho_\alpha$ such that $|OPT_\beta|$ is a constant fraction of $|OPT_\alpha|$.

Let $\sigma_k = [q_k, q'_k]$ be the kth label of OPT_α in the order from left to right. We associate A_k to σ_k, where A_k is the subset of labels of points in P at resolution α such that their right endpoints lie inside interval $[q'_k, q'_{k+1})$. Note that A_k includes σ_k itself. For convenience, we set $q'_{|OPT_\alpha|+1} = \infty$. For each label $\sigma = [p, p']$ in A_k, observe that $\rho_k(\sigma) = (q'_{k+1} - p)/(p' - p)$ is the smallest scaling ratio for σ to intersect q'_{k+1}. Thus $\rho_k = \max_{\sigma \in A_k} \rho_k(\sigma)$ for A_k is the smallest scaling ratio such that all labels in A_k intersect q'_{k+1}. Note that as $\sigma_k \in A_k$, $\rho_k(\sigma) \leq \rho_k$. We call the label of A_k which constitutes ρ_k the *dominating label* of A_k. Then it is clear that the following observation holds.

Lemma 2. *Consider A_k associated with a label $\sigma_k = [q_k, q'_k] \in OPT_\alpha$. If $\Delta_k = [\delta_k, \delta'_k]$ is the dominating label of A_k, then $q_k \leq \delta_k$.*

The above observation says that the dominating label of A_k has its left endpoint in the right of q_k. We set $\rho\,(= \rho_\beta/\rho_\alpha)$ to be the median value of all the ρ_k's. Then we claim that there are constants c_1 and c_2 such that $c_1|OPT_\alpha| \leq |OPT_\beta| \leq c_2|OPT_\alpha|$ as stated in the following lemma. Remark that we will assume all ρ_k's are different for simplicity to convey our idea. In fact, if some ρ_k's are the same, the following arguments still hold although the constants would deviate slightly.

Lemma 3. $\frac{1}{4}|OPT_\alpha| \leq |OPT_\beta| \leq \frac{3}{4}|OPT_\alpha|$.

Proof. We first prove the former part of the inequality. Let $L = \{A_k | \rho_k > \rho\}$, where we suppose the elements of L are ordered from left to right. Then $|L| = |OPT_\alpha|/2$ as by definition ρ is the median value of all the ρ_k. Note that at most one label from $A_k \in L$ can be selected for any labeling. Now we claim that the dominating labels of every other sets A_k in L do not overlap. Suppose A_i, A_j, A_k $(i < j < k)$ be three consecutive sets in L. Let $\Delta_i = [\delta_i, \delta'_i]$ and $\Delta_k = [\delta_k, \delta'_k]$ be the dominating labels of A_i and A_k respectively. At resolution

β, Let $\Delta_{i,\beta} = [\delta_i, \delta'_{i,\beta}]$ and $\Delta_{k,\beta} = [\delta_k, \delta'_{k,\beta}]$ be the scaled labels of Δ_i and Δ_k at resolution β respectively. We claim that $\Delta_{i,\beta}$ does not intersect $\Delta_{k,\beta}$. It suffice for us to prove $\delta'_{i,\beta} < \delta_k$. As $A_i \in L$, $\delta_{i,\beta} \leq q'_{i+1} \leq q'_j$. On the other hand, by Lemma 2, $q_k \leq \delta_k$ as Δ_k is dominating A_k. Also it is clear $q'_j < q_{j+1} \leq q_k$. Thus we have $\delta_{i,\beta} < q_k$. This implies if we select all the dominating labels of every other sets A_k in L from left to right, they cannot overlap. Hence $|OPT_\beta| \geq |L|/2 \geq \frac{1}{4}|OPT_\alpha|$.

We then prove the latter part of the inequality. Let $S = \{A_k | \rho_k \leq \rho\}$. Then $|S| = |OPT_\alpha|/2$, and $|OPT_\alpha| = |S| + |L|$. Let us also divide the OPT_β into two subsets L' and S' where $L' = \{\sigma \in OPT_\beta \mid \sigma \in A_k \text{ for some } A_k \in L\}$, and $S' = \{\sigma \in OPT_\beta \mid \sigma \in A_k \text{ for some } A_k \in S\}$. Then $|OPT_\beta| = |L'| + |S'|$. As a label σ in $A_k \in S'$ must overlap q'_{k+1}, which means that σ overlaps all the labels in A_{k+1}. Thus if a label σ in $A_k \in S'$ lies in OPT_β, then no labels in A_{k+1} can lie in OPT_β. We call A_{k+1} is *abandoned* when σ is selected in OPT_β. Now we put all the abandoned sets A_{k+1} (due to those labels from all the A_k in S' selected into OPT_β) into sets L_a or S_a such that $L_a \subset L$ and $S_a \subset S$. Then we have that $|S'| \leq |L_a| + |S_a|$ as each $A_k \in S$ can contribute at most one label in OPT_β. Also it holds that $|L'| + |L_a| \leq |L| = |OPT_\alpha|/2$ and $|S'| + |S_a| \leq |S| = |OPT_\alpha|/2$. If $|S_a| \geq |OPT_\alpha|/4$, then $|OPT_\beta| = |L'| + |S'| \leq |L'| + (|S| - |S_a|) \leq |L| + (|S| - |S_a|) \leq |OPT_\alpha|/2 + |OPT_\alpha|/4 = \frac{3}{4}|OPT_\alpha|$. Otherwise when $|S_a| < |OPT_\alpha|/4$, $|OPT_\beta| = |L'| + |S'| \leq |L'| + (|L_a| + |S_a|) \leq |L| + |S_a| \leq \frac{3}{4}|OPT_\alpha|$. ⊓⊔

Building the whole hierarchy. Build the lowest level takes $O(n \log n)$ time. We then construct the levels one by one upwards. To construct one level higher with resolution β from the level with resolution α, we need to first determine the scaling factor $\rho = \rho_\beta/\rho_\alpha$ as described previously so that $\frac{1}{4}|OPT_\alpha| \leq |OPT_\beta| \leq \frac{3}{4}|OPT_\alpha|$. This takes $O(n \log n)$ time. Then we can construct OPT_β in time $n \log |OPT_\alpha|$ by using *ComputeOPT*(P). We compute levels upwards until we reach a level at which the size of the optimal solution is a constant, and we stop. This gives us a $O(\log n)$-height hierarchy. It needs $O(n \log^2 n)$ time and $O(n)$ space in total. We summarize these in the following theorem.

Theorem 1. *A hierarchy of height $O(\log n)$ for the adaptive zooming query problem in one dimension can be built in time $O(n \log^2 n)$ using $O(n)$ space.*

2.3 Adaptive querying

With the low-height hierarchy, it is possible for us to answer zooming queries efficiently at any resolution.

Theorem 2. *Given a zooming query Q with window $W = [q, q']$ at resolution γ. Let OPT_γ be the optimal set of non-overlapping labels for points in P at resolution γ by running the greedy algorithm. Then with the $O(\log n)$-height hierarchy, the optimal solution for Q can be computed in $O(|\Phi_\gamma^W| \log(|OPT_\gamma|) + \log\log n)$ time, where Φ_γ^W is the set of labels intersecting the window W at resolution γ.*

Proof. First by binary search, use ρ_γ to locate the consecutive levels of resolutions α and β (where $\rho_\alpha < \rho_\gamma < \rho_\beta$) in the hierarchy. As the height of the hierarchy is $O(\log n)$, the location is done in $O(\log\log n)$ time.

Then we search for q and q', the endpoints of W. Suppose $q'_1, q'_2, \ldots, q'_{|OPT_\alpha|}$ be the right endpoints of the greedy solution at resolution α. We need to locate q and q' in these right endpoints. This needs $O(\log|OPT_\alpha|) = O(\log|OPT_\gamma|)$ time. Suppose q lies in $[q'_i, q'_{i+1}]$ and q' lies in $[q'_j, q'_{j+1}]$.

For each label whose right endpoints lying inside $[q'_k, q'_{k+1}]$ (where $i+1 \leq k \leq j$), check whether it completely lies inside W. So we can collect the set Ξ_γ^W of all the labels completely lying inside W at resolution γ. This needs $O(|\Phi_\gamma^W|)$ time.

Finally we use Ξ_γ^W to compute the optimal set of non-overlapping labels in W at resolution γ by using $ComputeOPT(\Xi_\gamma^W)$. This takes $O(|\Xi_\gamma^W|\log|OPT_\alpha|)$ $= O(|\Phi_\gamma^W|\log|OPT_\gamma|)$ time. ⊓⊔

3 Adaptive zooming in two dimensions

We then extend our idea to build the low-height hierarchy in two dimensions for efficient adaptive zooming queries. At each level of the hierarchy, we store the stabbing line structures as used in [1, 4]. This helps us build the hierarchy and efficiently answer adaptive zooming queries. Let OPT_γ denote the optimal solution the labels of P at resolution γ.

Suppose all the labels have unit height at the current resolution α. We suppose a label does not include its lower boundary for convenience. We can stab all the labels by a set of horizontal lines $\ell_1, \ell_2, \ldots, \ell_k$, ordered from top to bottom, satisfying three conditions: (i) each ℓ_i must stab at least one label; (ii) a label must intersect exactly one stabbing line; and (iii) two consecutive stabbing lines are separated with distance at least one. Let A_i be the set of labels stabbed by ℓ_i, and let $OPT_\alpha(A_i)$ be optimal labeling for labels A_i by running the one dimensional greedy algorithm on $\{\sigma \cap \ell_i | \sigma \in A_i\}$. We define the *stabbing line structure* $\mathcal{L}_\alpha$ at resolution α to be the set all the stabbing lines ℓ_i at resolution α together with A_i and $OPT_\alpha(A_i)$.

Unlike in one dimension, OPT_α cannot be derived easily from the stabbing line data structure $\mathcal{L}_\alpha$ in two dimensions. However, a 2-approximation to OPT_α can be obtained easily from $\mathcal{L}_\alpha$. Let X_α^{odd} (resp., X_α^{even}) be the union of $OPT_\alpha(A_i)$ for odd i (resp., for even i). As any pair of consecutive stabbing lines is separated by a distance at least one, the labels in X_α^{odd} never overlap those in X_α^{even} and vice versa. Thus if we take the maximum-size labeling of $OPT(X_\alpha^{\text{odd}})$ and $OPT(X_\alpha^{\text{even}})$, it is a 2-approximation for OPT_α. Moreover, $\mathcal{L}_\alpha$ can help us find the $(1+\epsilon)$-approximation [1, 4]. We then describe a way to find $\mathcal{L}_\beta$ at resolution β (which is coarser than α) by making use of $\mathcal{L}_\alpha$. This will serve as the main subroutine to build the hierarchy.

3.1 Computing $\mathcal{L}_\beta$ from $\mathcal{L}_\alpha$

For convenience, we assume $\rho_\alpha = 1$, $\rho_\beta = \rho$, and labels at resolution α has unit height. Let ℓ be any horizontal line at resolution β with y-coordinate $y(\ell)$. Let B_ℓ be the set of labels that intersect ℓ at resolution β. Let H be the horizontal strip bounded by the horizontal lines at $y(\ell) - \rho$ and at $y(\ell) + \rho$. Suppose that $\{\ell_i, \ldots, \ell_j\}(i < j)$ is the set of the stabbing lines at resolution α lying inside H. Observe that the labels in B_ℓ can only be members of $A_i, \ldots, A_j$ at resolution α. Now suppose that S_ℓ is the ordered sequence of the right endpoints of the intervals in $OPT_\alpha(A_i), \ldots, OPT_\alpha(A_j)$ projected onto ℓ. Then we can obtain the optimal labeling $OPT_\beta(B_\ell)$ for the labels in B_ℓ by executing $ComputeOPT(B_\ell)$ using the points in S_ℓ as separators to partition the labels in groups. This takes $O(|B_\ell| \log |S_\ell|)$ time.

Now we describe how we draw the stabbing lines $\ell'_1, \ell'_2, \ldots$ from top to bottom to stab all the labels at resolution β. First, we draw the first line $\ell'_1 = \ell_1$, and collect $B_{\ell'_1}$ and compute $OPT_\beta(B_{\ell'_1})$ as described in the previous paragraph. We then draw the second stabbing line ℓ'_2 with y-coordinate $y(\ell'_1) - \rho$ if it intersects some labels at resolution β. Otherwise we set ℓ'_2 to be the stabbing line ℓ_i below and nearest to the y-coordinate $y(\ell'_1) - \rho$. We continue this process until all labels at resolution β are stabbed. Note that the right endpoints of labels in $OPT_\alpha(A_i)$ at resolution α may be used twice to compute $OPT_\beta(B_\ell)$ for two consecutive stabbing lines at resolution β. In total, it takes $O(n \log |OPT_\alpha|)$ time to compute $\mathcal{L}_\beta$. We summarize the result as the following lemma.

Lemma 4. *Given $\mathcal{L}_\alpha$ at resolution α. Then $\mathcal{L}_\beta$ at a coarser resolution β can be computed from $\mathcal{L}_\alpha$ in $O(n \log |OPT_\alpha|)$ time.*

3.2 Building $O(\log n)$-height hierarchy

Building the lowest level. We build the lowest level at which the labels at distinct points do not overlap. By considering the projections of the labels onto x- and y-axes respectively, it is not hard to decide a resolution such that for each pair of points, either the x-projections or the y-projections of their labels do not overlap. This can be done in $O(n \log n)$ time.

Building one level higher. We have known how to construct the stabbing line structure $\mathcal{L}_\beta$ for a resolution β by making use of $\mathcal{L}_\alpha$ at a finer resolution α if the scaling factor $\rho = \rho_\beta / \rho_\alpha$ is known. In order to have a low-height hierarchy, it suffices for us to find a scaling factor ρ such that $|OPT_\beta|$ is a constant fraction of $|OPT_\alpha|$. For convenience, we assume that the height of labels at resolution α is unit.

At resolution α, a set A_ℓ of labels that intersect a stabbing line ℓ is partitioned into several groups by labels in $OPT_\alpha(A_\ell)$ (as in the one-dimensional case). Each of the groups consists of labels whose right endpoints lie between the right endpoints of two consecutive labels in $OPT_\alpha(A_\ell)$. The intersection of labels in such a group in A_ℓ is called a *kernel* (denoted by K_α), and those labels in that group are said to be *associated* with the kernel K_α. Let $\mathcal{K}_\alpha^{\text{odd}}$ and $\mathcal{K}_\alpha^{\text{even}}$ be the collections of all the kernels intersecting odd and even stabbing lines at resolution α, respectively. Let $\mathcal{K}_\alpha = \mathcal{K}_\alpha^{\text{odd}} \cup \mathcal{K}_\alpha^{\text{even}}$. Let A_{odd} and A_{even} be the set of all labels intersecting odd and even stabbing lines at resolution α, respectively. We then use the interactions of the scaled versions of the kernels in $\mathcal{K}_\alpha$ to decide the scaling factor ρ.

The labels at resolution β are obtained by scaling the labels of resolution α by factor ρ. The kernels in $\mathcal{K}_\alpha^{\text{odd}}$ and $\mathcal{K}_\alpha^{\text{even}}$ are enlarged to the kernels at resolution β and we denote the corresponding sets of enlarged kernels at resolution β simply by $\mathcal{K}_{\text{odd}}$ and $\mathcal{K}_{\text{even}}$ respectively. Let $\mathcal{K} = \mathcal{K}_{\text{odd}} \cup \mathcal{K}_{\text{even}}$. The labels in A_{odd} and A_{even} become B_{odd} and B_{even} respectively. We also denote the scaled version of kernel K_α by K. Two kernels (resp. labels) are said to be of the *same parity* if they are contained in the same kernel (resp. label) collection $\mathcal{K}_{\text{odd}}$ or $\mathcal{K}_{\text{even}}$ (resp. A_{odd} or A_{even}). For each kernel K_α at resolution α, scale it until it intersects the left sides of first three other kernels of the same parity. We denote this scaling ratio for K_α by $\rho(K_\alpha)$. We set the scaling factor ρ to be the $(\frac{10|\mathcal{K}_\alpha|}{11})$-th smallest value of $\rho(K_\alpha)$ for all kernels $K_\alpha \in \mathcal{K}_\alpha$. With this ratio ρ, we claim that the optimal labeling at resolution β is a constant fraction of that at resolution α. Note that as the one-dimensional case, we will assume all the $\rho(K_\alpha)$ are distinct for convenience to convey our idea.

For a kernel K, let $R(K)$ be a label whose right side constitutes the right side of K. We call $R(K)$ the *right representative* of K. We denote the height and width of a kernel or label by $\tau(\cdot)$ and $\omega(\cdot)$ respectively. Also we denote by $x_l(K)$ and $x_r(K)$ the x-coordinates of the left and right sides of a kernel K respectively. First of all, the following two observations are clear.

Lemma 5. *Suppose $K \in \mathcal{K}$ at resolution β is obtained by scaling ρ times a kernel K_α at resolution α. Then its height $\tau(K_\alpha) \leq \tau(K) \leq \tau(K_\alpha) + (\rho - 1)$, and its width $\omega(K) \geq \omega(K_\alpha) + \left(\frac{\rho-1}{\rho}\right)\omega(R(K))$. Moreover if $R(K)$ is different from $R(K_\alpha)$, then $x_l(R(K)) > x_l(R(K_\alpha))$.*

Lemma 6. *Let J, K be two non-intersecting kernels on the same stabbing line ordered from left to right at resolution β, where J, K are obtained by scaling ρ times the kernels J_α, K_α at resolution α respectively. Then $x_r(R(J)) < x_r(K)$ and $x_l(J) < x_l(R(K))$.*

Let S_{odd} (resp. L_{odd}) be the subset of kernels K in $\mathcal{K}_{\text{odd}}$ with $\rho(K_\alpha)$ smaller (resp. not smaller) than ρ. Similarly, we define S_{even} and L_{even}. Let $S = S_{\text{odd}} \cup S_{\text{even}}$ and $L = L_{\text{odd}} \cup L_{\text{even}}$. The following lemma tells us that there is a large set of non-intersecting kernels in L_{odd} or L_{even}.

Lemma 7. *For any $i \in \{\text{odd}, \text{even}\}$, each kernel in L_i can intersect at most $1.5\rho + 12$ kernels in L_i.*

Proof. Let K be any kernel in L_i. For a kernel J in L_i intersecting K, we put it into I_1 if $x_l(J) \leq x_l(K)$ and into I_2 otherwise. As K intersects the left sides of all kernels in I_2, $|I_2| \leq 3$.

To determine $|I_1|$, we divide I_1 into three subsets depending on whether the kernels in I_1 intersect the supporting lines ℓ^t and ℓ^b of the top and bottom sides of K. If those kernels intersects ℓ^t (resp., ℓ^b), then put them into I_1^t (resp. I_1^b). Otherwise, i.e., if they lie between ℓ^t and ℓ^b, they are put into I_1^m. We first claim that $|I_1^t| \leq 3$. Suppose for the contradiction that $|I_1^t| \geq 4$. All the kernels in I_1^t must contain the top-left corner of K. This means that the kernel J in I_1^t with the smallest $x_l(J)$ would intersect the left sides of at least four kernels of same parity. This implies that $J \notin L_i$, which is a contradiction. Thus $|I_1^t| \leq 3$. A similar argument proves $|I_1^b| \leq 3$.

Then we bound $|I_1^m|$. As the height of K is at most ρ. There are at most $\frac{\rho}{2}+1$ stabbing lines with the same parity between ℓ^t and ℓ^b. As on any of these stabbing line, there are at most three non-intersecting kernels in L_i stabbed before K, $|I_1^m| \leq 3(\frac{\rho}{2}+1) = 1.5\rho + 3$.

Considering all together, $|I_1| + |I_2| \leq 1.5\rho + 12$. ⊔⊓

Let N_i be any maximal subset of non-intersecting kernels in L_i for $i \in \{\text{odd}, \text{even}\}$. By Lemma 7, we have $|N_i| \geq |L_i|/(1.5\rho + 13)$. Although no two kernels in N_i intersect each other, their right representatives may intersect. The following lemma proves that the number of those right representatives which intersect $R(K)$ for a kernel $K \in N_i$ is bounded above by $O(\rho)$. The argument is similar to Lemma 7 by packing kernels and labels around $R(K)$. This in turn implies that there are at least $\Omega(|L_i|)$ non-intersecting labels in B_i as stated in Lemma 9.

Lemma 8. *Let $K \in N_i$ be the kernel with its right representative label $R(K)$ of the shortest width among all other kernels in N_i. Then $R(K)$ can intersect the right representative labels of less than $6\rho + 4$ kernels in N_i.*

Lemma 9. *There are at least $\frac{|L_i|}{(3\rho+2)(3\rho+26)}$ non-intersecting labels in B_i for any $i \in \{\text{odd}, \text{even}\}$.*

Now we are well-equipped to show the main lemma, whose proof uses the similar idea as Lemma 3 in one dimensional case.

Lemma 10. *There exist constants $0 < c_1, c_2 < 1$ such that $c_1|OPT_\alpha| \leq |OPT_\beta| \leq c_2|OPT_\alpha|$.*

Note that the details of the proofs of Lemma 8, 9 and 10 are omitted in this preliminary version. We then describe the algorithm to compute the scaling factor ρ, and analyze its running time.

For each kernel K_α, $\rho(K_\alpha, K'_\alpha)$ can be determined in $|K_\alpha| \cdot |K'_\alpha|$ time, where $|K_\alpha|$ is the number of the associated labels of K_α. For a fixed K_α, to compute all $\rho(K_\alpha, K'_\alpha)$ for different K'_α, it takes $n|K_\alpha|$ time. To determine the third smallest value $\rho(K_\alpha)$ out of all $\rho(K_\alpha, K'_\alpha)$, it requires at most $3|\mathcal{K}_\alpha|$ time. In total, to determine $\rho(K_\alpha)$, it takes at most $n(|K_\alpha| + 3)$ time.

To determine all $\rho(K_\alpha)$ for all K_α, it takes time $n(|K_\alpha|+3)$ summing over all kernels $K_\alpha \in \mathcal{K}_\alpha$. This takes time $O(n^2)$ to determine all $\rho(K_\alpha)$. Furthermore, the $\frac{10}{11}m$-th value ρ of all $\rho(K_\alpha)$ can be determined in $O(n \log n)$ time. In all, ρ can be computed in $O(n^2)$ time.

Auxiliary structures for efficient querying. In order to efficiently locate all the labels intersecting a specific window, we associate a range tree R_α with each hierarchy level say at resolution α. We collect all the intersection points S of the boundaries of all kernels with all the stabbing lines. We then build a 2-dimensional range tree R_α on the point set S. This takes $O(|S| \log(|S|)) = O(|OPT_\alpha| \log |OPT_\alpha|)$ time and space [6]. The query time to report all the points from S inside a rectangular window W is $O(\log(|S|)+k) = O(\log(|OPT_\alpha|)+k)$ where k is the number of points inside W. Remark that we suppose the technique of fractional cascading is applied to the range tree; otherwise, the query time can go up to $O(\log^2(|S|) + k)$.

Building the whole hierarchy. Combining the above statements, we have the following theorem.

Theorem 3. *A hierarchy of height $O(\log n)$ for the adaptive zooming query problem in two dimensions can be built in $O(n^2 \log n)$ time using $O(n \log n)$ space.*

3.3 Adaptive querying

Theorem 4. *Given any zooming query Q with window $W = [x, x'] \times [y, y']$ at resolution γ. Let OPT_γ be the optimal set of non-overlapping labels for points in P at resolution γ. Let Φ_γ^W be the set of labels intersecting the window W at resolution γ. Suppose the $O(\log n)$-height hierarchy is given. Then*

(i) The 2-approximation for Q can be computed in $O(|\Phi_\gamma^W| \log(|OPT_\gamma|) + \log \log n)$ time.

(ii) The $(1+\epsilon)$-approximation for Q can be computed in $O(|\Phi_\gamma^W|^{\frac{1}{\epsilon}} + \log(|OPT_\gamma|) + \log \log n)$ time.

Proof. First by binary search, use ρ_γ to locate the consecutive levels of resolutions α and β where $\rho_\alpha < \rho_\gamma < \rho_\beta$ in the hierarchy. As the height of the hierarchy is $O(\log n)$, the location is done in $O(\log \log n)$ time.

Search the auxiliary range tree R_α at resolution α to find all points lying inside W. This takes $O(\log(|OPT_\alpha|) + k)$ time, where k is the number of points of R_α inside W. Note that $\log(|OPT_\alpha|) = O(\log(|OPT_\gamma|))$. As the labels intersecting W at resolution α will continue intersecting W at resolution γ, we have $k = O(|\Phi_\gamma^W|)$. For groups of labels corresponding to these k kernels, we check one by one whether they are inside W or not. So we can collect all the labels Ξ_γ^W completely lying inside W at resolution γ in $O(|\Phi_\gamma^W|)$ time.

(i) We use Ξ_γ^W to compute the 2-approximation solution for Q by applying the one-dimensional greedy algorithm on related stabbing lines. This takes $O(|\Xi_\gamma^W| \log |OPT_\alpha|) = O(|\Phi_\gamma^W| \log |OPT_\gamma|)$ time.

(ii) We use Ξ_γ^W to compute the $(1+\epsilon)$-approximation for Q by applying the algorithm by Chan [4]. This takes $O(|\Xi_\gamma^W|^{\frac{1}{\epsilon}}) = O(|\Phi_\gamma^W|^{\frac{1}{\epsilon}})$ time. ◻

4 Conclusion & Discussion

In this paper, we build low-height hierarchies for one and two dimensions for answering adaptive zooming queries efficiently. In the model we have considered, the labels at any point are restricted to several fixed positions lying on the right of the point. One interesting research direction is to extend our results to point labeling for more general models, for example sliding models. Can some notion of point importance be added into the data structure? Can we build a hierarchy for a subdivision map to help us query the map area in a window at any resolution?

References

1. P.K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, vol. 11, pp. 209-218, 1998.
2. M. Arikawa, Y. Kambayashi, and H. Kai. Adaptive geographic information media using display agents for name placement. *Trans. Journal of the Geographic Information Systems Association*, 1997. in Japanese.
3. M. Arikawa, H. Kawakita, and Y. Kambayashi. An environment for generating interactive maps with compromises between users' requirements and limitations of display screens. *Trans. Journal of the Geographic Information Systems Association*, 2, March 1993. in Japanese.
4. T. Chan. A Note on Maximum Independent Sets in Rectangle Intersection Graphs. *Information Processing Letters*, 89(1), 19–23, 2004.
5. B. Chazelle and 36 co-authors. The computational geometry impact task force report. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, vol. 223, pages 407-463. American Mathematical Society, Providence, 1999.
6. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Heidelberg, 1997.
7. M. Formann and F. Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 281–288, 1991.
8. D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM J. Discrete Math.*, 5(3), 422–427, 1992.
9. MapFactory. http://home.coqui.net/rjvaasjo/index.htm
10. J. Marks and S. Shieber. The computational complexity of cartographic label placement. Technical report, Harvard CS, 1991.
11. I. Petzold, L. Plumer, and M. Heber. Label placement for dynamically generated screen maps. In *Proceedings of the Ottawa ICA*, pp. 893-903, 1999.
12. S. Roy, P.P. Goswami, S. Das, and S.C. Nandy. Optimal Algorithm for a Special Point-Labeling Problem In *Proc. 8th Scandinavian Workshop on Algorithm Theory (SWAT'02)*, pp. 110-120, 2002.
13. T. Strijk and A. Wolff. www.math-inf.uni-greifswald.de/map-labeling/bibliography/
14. M. van Kreveld. http://www.cs.uu.nl/centers/give/geometry/autocarto/index.html
15. M. van Kreveld, T. Strijk, and A. Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, vol. 13, pp. 21-47, 1999.
16. M. van Kreveld, R. van Oostrum, and J. Snoeyink. Efficient settlement selection for interactive display. In *Proc. Auto-Carto 13: ACSM/ASPRS Annual Convention Technical Papers*, pp. 287-296, 1997.